Approximation algorithms to the network design problems(Dissertation_全文)

AUTHOR(S):

Fukunaga, Takuro

Approximation Algorithms to the Network Design Problems

Takuro Fukunaga

Department of Applied Mathematics and Physics
Graduate School of Informatics
Kyoto University
Kyoto 606-8501, Japan

January, 2007

Doctoral Dissertation

submitted to Graduate School of Informatics, Kyoto University

in partial fulfillment of the requirement for the degree of

DOCTOR OF INFORMATICS

(Applied Mathematics and Physics)

Preface

In spite of the outstanding progress in combinatorial optimization, most problems classified as NP-hard were left untouched for a long time. However, some challenges have appeared for dealing with such problems recently. One of them is the framework of approximation algorithms. An approximation algorithm usually guarantees that the cost of the computed solution is bounded by some factor times the optimal cost from the the above (resp., below) in a minimization (resp., maximization) problem. In approximation algorithms, the factors are regarded as important, as well as their running time. This criterion provides us with a new point of view about the difficulty of optimization problems, especially in NP-hardness.

In this thesis, we treat combinatorial optimization problems related to the network design, and examine the structures and algorithms of them from the view point of the approximability. Most of the network design can be roughly described as follows; Given a graph cost and some required conditions, construct a minimum cost graph that satisfies the conditions. Such problems are important since they provide flexible frameworks. Actually they contain a variety of problems, which are theoretically and practically valuable.

In concrete, we first consider several network design problems with the edge dominating set constraints. This constraint bounds the number of edges that dominate some edge. We investigate the approximability of these problems. Approximation guarantees of our algorithms are derived from the relationship between polyhedra defined from the problems.

Next, we consider problems with the edge-connectivity. Since it represents a tolerance of networks for deficit of edges, the edge-connectivity is one of the central topics in the network design. We consider problems with the edge-connectivity requirement and degree bounds, and reveal several conditions for them to admit constant-factor approximation algorithms. For this, we investigate graph transformations called splitting and detachment. These transformations present useful tools for developing algorithms to the network design with the edge-connectivity.

Finally, we extend the edge-connectivity between two vertices to between subsets of vertices. We then consider a problem that asks to construct a least cost graph connecting given families of vertex subsets. Our approximation algorithm for this problem is based on the approximate integer decomposition property, which can be derived from a packing theorem.

We believe that our algorithms for the above problems are useful and developed techniques give a new insight into network design problems. We hope that the works in this thesis will be helpful to advance the study in these topics.

January, 2007
Takuro Fukunaga

Acknowledgment

Without support and encouragement from numerous people, I could have never completed this work. I would like to acknowledge them here.

First of all, I would like to express my sincere appreciation to Professor Hiroshi Nagamochi of Kyoto University for his guidance. He gave me much opportunity and continuous support for studying this area. His advice provided insights into research activity for me.

I am deeply grateful to Professor Toshihide Ibaraki of Kwansei Gakuin University. He welcomed me to his laboratory and supervised my bachelor's thesis. He let me notice fascinating aspect of this field.

I wish to express my gratitude to Professor Mutsunori Yagiura of Nagoya University, Professor Koji Nonobe of Hosei University, Professor Liang Zhao of Kyoto University, and all members in Discrete Mathematics laboratory of Kyoto University for their warm friendship in the period I studied at this laboratory.

Professor András Frank and Professor Tibor Jordán of Eötvös Lorlánd University, and members of their research group hosted my visit from September to December 2005. Professor Ojas Parekh of Emory University commented on results described in Chapter 5. I would like to thank them.

I am also thankful to Professor Masao Fukushima of Kyoto University and Professor Kazuo Iwama of Kyoto University for serving on my dissertation committee.

Finally, I would like to express my gratitude to my family for their heartfelt cooperation and encouragement.

Contents

List of Figures

List of Problems

Chapter 1

Introduction

1.1 Background

An instance of the *optimization problem* consists of an objective function and constraints. A solution is called *feasible* if it satisfies given constraints. The problem asks to find an *optimal solution*, which is feasible and minimizes (or maximizes) the objective function. The *combinatorial optimization problem* especially deals with objects that have combinatorial structures in some sense.

If the number of feasible solutions is small, we can find an optimal solution in a combinatorial optimization problem by searching all feasible solutions. However, this cannot be a choice when the number is large (particularly infinite) because the search may spend huge amount of running time. In this area, the most popular criterion for efficient algorithms is *polynomial time solvability*, proposed by J. Edmonds in the 1960s. This advocates that an algorithm is regarded as efficient if its running time is always bounded by a polynomial of the size of a given instance. Under this criterion, efficient algorithms for a variety of combinatorial optimization problems were considered and examining them leads to outstanding advance of this field. However, problems classified as *NP-hard* were left untouched for a long time, where NP-hard is the class of problems widely believed to admit no efficient (i.e., polynomial time) algorithms.

Recently, some kinds of challenges for dealing with NP-hard problems have been extensively considered. One of them is the framework of *approximation algorithms*. An approximation algorithm usually guarantees that the cost of the computed solution is bounded by some factor times the optimal cost from the above (resp., below) in a minimization (resp., maximization) problem. In approximation algorithms, the factors in the approximation guarantee are regarded as important, as well as their running time. This criterion provides us with a new point of view about the difficulty of problems, especially in NP-hardness. Moreover, designing approximation algorithms has generated new useful techniques for solving combinatorial optimization problems. For example, rounding a solution of a linear programming into an integer solution is one of the basic methods to design approximation algorithms, and it has made the insight into the integrality of polyhedra deeper than before.

In this thesis, we treat combinatorial optimization problems related to the network design, and examine the structures and algorithms of them from the view point of the approximability.

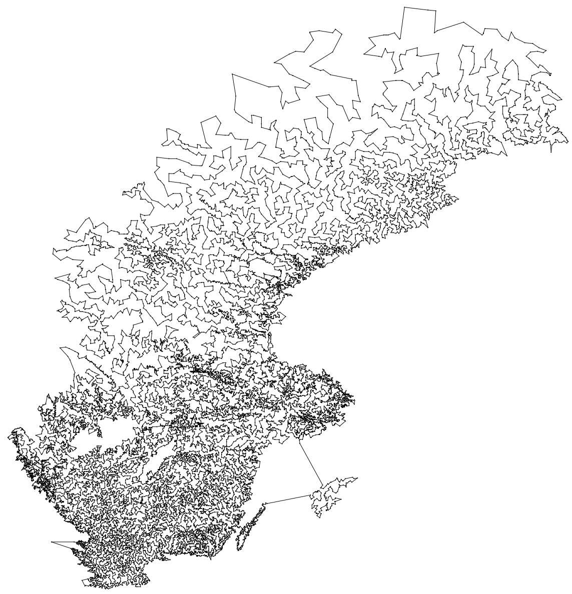


Figure 1.1: An optimal solution of TSP for an instance constructed from cities of Sweden

Most of the network design can be roughly described as follows; Given a graph cost and some required conditions, construct a minimum cost graph that satisfies the conditions. Such problems are important since they provide a flexible framework. Actually they contain a variety of problems, which are theoretically and practically valuable. In what follows, we see the *traveling salesperson problem* (TSP) and *Steiner tree problem* as representative examples of them, mentioning their interesting aspects.

First, let us see the TSP. An edge set H is called a *Hamiltonian cycle* on a vertex set V if $G_H = (V, H)$ is connected and the degree of each vertex in V is exactly two. Let $\mathbb{Q}_+$ stand for the set of non-negative rationals. Then TSP is defined as follows.

Traveling salesperson problem (TSP)
Given a vertex set V and an edge cost $w : \binom{V}{2} \to \mathbb{Q}_+$, find a minimum cost Hamiltonian cycle on V.

We note that the TSP is known to be NP-hard [52]. Figure 1.1 illustrates an optimal solution computed by an algorithm due to D. Applegate et al. [2] for an instance constructed from 24978 cities in Sweden.

The TSP is one of the most extensively examined problems in the combinatorial optimization. It has yielded much progress of theories of computational complexity and algorithms. Actually many useful techniques such as the branch and bound, cutting plane, and local search have been developed stimulated by solving the TSP.

We can immediately imagine an application of this problem to deciding routings of some objects, say, plains, trains and vehicles. In addition to this, there are various applications because of its simpleness. As an example, let us introduce a scheduling problem of a single machine.

Scheduling a single machine
Consider processing jobs $j_1, \ldots, j_n$ by a single machine, where set-up time $w(j_{i'} j_i) \in \mathbb{Q}_+$ is given for processing j_i after $j_{i'}$. Find an order of the jobs minimizing the completion time of all jobs.

This scheduling problem can be reduced to the TSP as follows. Prepare a dummy job j_0 with set-up time $w(j_0 j_i) = w(j_i j_0) = 0$ for $i = 1, \ldots, n$, and let $V = \{j_0, \ldots, j_n\}$ and w be an instance of the TSP. Then we can obtain a permutation $(i_1, \ldots, i_n)$ of $(1, \ldots, n)$ from an optimal solution $H = \{j_0 j_{i_1}, j_{i_1} j_{i_2}, \ldots, j_{i_{n-1}} j_{i_n}\}$ for the instance. This permutation stands for an optimal order of jobs.

Next, we introduce the Steiner tree problem. We say that two vertices u and v are *connected* in a graph if there exists a path whose end vertices are u and v.

Steiner tree problem
Given an undirected graph $G = (V, E)$, an edge cost $w : E \to \mathbb{Q}_+$, and set $X \subseteq V$ of terminals, find a minimum cost tree $T \subseteq E$ connecting terminals in X.

Every feasible solution of the Steiner tree problem is called a *Steiner tree*. Figure 1.2 illustrates a Steiner tree denoted by dotted lines for a given graph and terminals denoted by black vertices.

Analogously to the TSP, the Steiner tree problem is also known to be NP-hard [28], and has diverse applications. For example, connecting terminals in VLSI with least length can be formulated as the Steiner tree problem. Many variants of this problem are also considered, which reflects its usefulness. In this thesis, some of the variants will appear in Chapter 6.

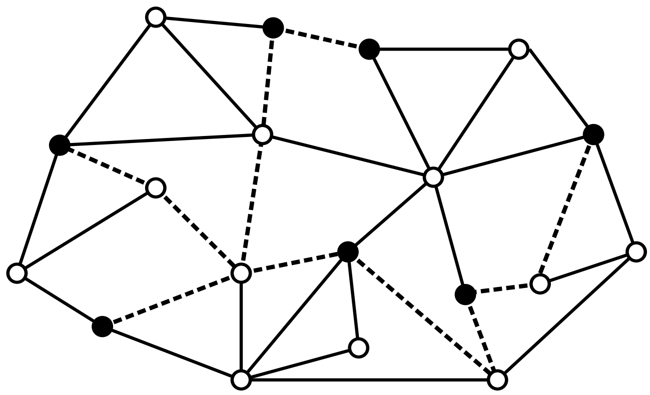


Figure 1.2: A Steiner tree denoted by dotted lines for a graph and terminals denoted by black circles

1.2 Organization of this thesis

This thesis consists of seven chapters including this introduction and the conclusion. In Chapter 2, we introduce notations and preliminary facts which we use in the subsequent chapters. Our main results are described in Chapters 3-6 which are summarized as follows.

Chapter 3: Network Design with Edge Dominating Constraints

We consider variants of the edge dominating set problem and the edge cover problem. Throughout Chapter 3, we derive the relationship between polyhedra, and utilize obtained properties to analyze approximation algorithms.

In the first problem we consider, the edge dominating set problem is generalized so that each edge in a graph has capacities on the number of edges dominating it and on the number of its multiple copies. We call this problem capacitated b-edge dominating set problem.

Next problem is a packing version of the capacitated b-edge dominating set problem. This can be regarded as a generalization of the induced matching problem. Hence we call it capacitated induced matching problem. In addition to proposing an approximation algorithm, we prove a result about the approximation hardness of the k-separated matching problem, which is another generalization of the induced matching problem.

Hyperedge dominating set problem finds a set of hyperedges dominating all the other hyperedges in a hypergraph. Generalized (a, c)-edge cover problem restricts the sums of degrees over some vertex subsets. We show that these are also approximated as in the above problems.

Since these problems have simple settings, there are many applications. Facility location problem, edge coloring problem, and marriage problem are examples of them.

Chapter 4: Splitting and Detachment with Local Edge-Connectivity

We consider graph transformations which are called splitting and detachment. Splitting is an operation that replaces two edges us and vs incident to a vertex s by a new edge uv. Detachment is an operation that splits every vertex v into some new vertices in a given vertex set V_v changing end vertices of each edge $uv \in E$ from u to some $x \in V_u$ and from v to some $y \in V_v$. The detachment is a generalization of the splitting. Although these transformations may decrease the edge-connectivity of graphs, conditions for these to preserve the edge-connectivity were proposed by several researchers. In Chapter 4, we derive new results on these transformations.

Concretely, we define a strongly splittable pair as a pair of edges incident to a vertex $s \in V$ such that splitting them preserves the local edge-connectivity between every two vertices in $V - s$ completely, and that between s and another up to the degree of s after the splitting. We show that there always exists a strongly splittable pair unless a cut-edge is incident to s or the degree of s is 3. Moreover, we show the existence of a splittable pair containing a designated edge in Eulerian graphs. Furthermore, we see that these results are extended to the detachments. Conditions for the detachments to generate no loop are also considered.

Results in Chapter 4 provide useful tools for dealing with the edge-connectivity of graphs. Actually, we use these to design algorithms in Chapter 5.

Chapter 5: Network Design with Edge-Connectivity and Degree Constraints

We consider a problem of constructing a minimum cost multigraph under an edge-connectivity requirement and degree bounds. Concretely, an instance of this problem consists of a vertex set V, an edge-connectivity requirement $r : \binom{V}{2} \to \mathbb{Z}_+$, a degree lower bound $a : V \to \mathbb{Z}_+$, a degree upper bound $b : V \to \mathbb{Z}_+$, and an edge cost $c : \binom{V}{2} \to \mathbb{Q}_+$, where $\mathbb{Z}_+$ stands for the set of non-negative integers. Then it asks to find a minimum cost multigraph $G = (V, E)$ such that the edge-connectivity between every two vertices u and v is at least $r(u, v)$ and the degree of a vertex v is at least $a(v)$ and at most $b(v)$. This problem provides a flexible framework. Indeed, it generalizes some fundamental problems such as the traveling salesperson problem and the vehicle routing problem. We reveal several conditions on the edge-connectivity requirements and degree bounds for which the problem admits a constant-factor approximation algorithm.

Chapter 6: The Set Connector Problem

Given a graph $G = (V, E)$ with an edge cost and families $\mathcal{V}_i \subseteq 2^V$, $i = 1, 2, \ldots, m$ of disjoint subsets, an edge subset $F \subseteq E$ is called a set connector if, for each $\mathcal{V}_i$, the graph $(V, F)/\mathcal{V}_i$ obtained from (V, F) by contracting each $X \in \mathcal{V}_i$ into a single vertex x has a property that every two contracted vertices x and x' are connected in $(V, F)/\mathcal{V}_i$. In Chapter 6, we introduce a problem of finding a minimum cost set connector, which contains several important network design problems such as the Steiner forest problem, the group Steiner tree problem, and the NA-connectivity augmentation problem as its special cases.

We derive an approximate integer decomposition property from a fractional packing theorem of set connectors, and present a 2α-approximation algorithm for the set connector problem, where $\alpha = \max_{1 \leq i \leq m}(\sum_{X \in \mathcal{V}_i} |X|) - 1$.

Chapter 2

Preliminaries

In this chapter, we introduce basic notations and facts used throughout this thesis. For standard definitions in graph theory and combinatorial optimization theory, refer to books on them, for example, [39, 51, 70, 71, 77].

2.1 Notations

Here we define notations.

2.1.1 Set, vector and function

We let $\mathbb{R}$, $\mathbb{Q}$ and $\mathbb{Z}$ stand for the sets of reals, rationals, and integers, respectively. Moreover, $\mathbb{R}_+$, $\mathbb{Q}_+$, $\mathbb{Z}_+$ and $\mathbb{Z}_+^{\mathrm{ev}}$ denote the sets of non-negative reals, non-negative rationals, non-negative integers and non-negative even integers, respectively. For an integer $k \in \mathbb{Z}_+$, θ_k denotes the k-th harmonic number, i.e., $\theta_k = \sum_{i=1}^{k} 1/i$.

For a set S, S^k denote the family of ordered multisets consisting of k elements in S. We sometimes consider an $|S|$-dimensional real vector $x \in \mathbb{R}^S$ whose entries correspond to elements of S. We represent an entry of x corresponding to an element s of S by $x(s)$. Since x can be also regarded as a function from S to $\mathbb{R}$, we often identify a vector with a function, and vice versa.

A support S_x of x is defined as the set of $s \in S$ for which $x(s) > 0$, i.e., $S_x = \{s \in S \mid x(s) > 0\}$. For a subset S' of S, we let $x(S') = \sum_{s \in S'} x(s)$, where we define $x(\emptyset) = 0$ for convenience. Moreover, $x_{S'} \in \mathbb{R}^{S'}$ denote the projection of x onto S'. We refer to the incidence vector of S' by $\mathcal{X}_{S'} \in \{0,1\}^S$, i.e., $\mathcal{X}_{S'}(s) = 1$ if $s \in S'$, and $\mathcal{X}_{S'}(s) = 0$ if $s \in S - S'$. If S' is a multi-subset of S, incidence vector $\mathcal{X}_{S'}$ of S' is in $\mathbb{Z}_+^S$. We sometimes use a subset (or multi-subset) and its incidence vector, interchangeably.

If $x(s) \leq \ell$ (resp., $x(s) \geq \ell$) holds for all $s \in S$ and some $\ell \in \mathbb{R}$, we say that $x \leq \ell$ (resp., $x \geq \ell$). As well as this, $x = \ell$ means that both of $x \leq \ell$ and $x \geq \ell$ hold.

2.1.2 Undirected graph

Let $G = (V, E)$ denote an undirected graph with vertex set V and edge set E, where an edge is defined as an unordered pair of vertices in V. We denote an edge consisting of vertices u

and v by uv. In this thesis, G is usually a multigraph , i.e., edge set contains some parallel edges, unless stated otherwise. We distinguish two parallel edges $e_1 = uv$ and $e_2 = uv$, which may be simply denoted by uv and uv.

For a vertex subset $U \subseteq V$, $G[U]$ denotes a subgraph of G induced by U, and $E[U]$ denotes the edge set of $G[U]$, i.e., the set of edges whose both end vertices are in U. For two vertex subsets $U, W \subseteq V$, $\delta(U, W; G)$ represents the set of edges in E whose one end vertex is in U and the other is in W. If U consists of only a single vertex v and there is no confusion, we represent U by v. Then $\delta(v, v; G)$ is the set of loops in E incident to vertex v. Moreover $\delta(U, V - U; G)$ is simply denoted by $\delta(U; G)$, where we let $\delta(V; G) = \emptyset$ for convenience. Furthermore we let $d(U, W; G) = |\delta(U, W; G)|$ and $d(U; G) = |\delta(U; G)|$. Note that the degree $\deg(v; G)$ of vertex v is defined as $2d(v, v; G) + d(v; G)$. If G contains no loop, we use $\deg(v; G)$ and $d(v; G)$ interchangably since $\deg(v; G) = d(v; G)$. A graph G is called *Eulerian* if $\deg(v; G)$ is even for all $v \in V$.

For a partition $\mathcal{P} = \{V_1, \ldots, V_p\}$ of V into non-empty subsets, $\delta(\mathcal{P}; G)$ denotes $\cup_{i=1}^{p} \delta(V_i; G)$. For an edge $e \in E$, $\delta(e; G)$ denotes the set of edges which share at least one end vertex with e, where $e \in \delta(e; G)$. When the graph under consideration is obvious, we may omit to clarify G in these notations.

For a vertex $v \in V$, $\Gamma(v)$ denotes the set of neighbors of v. For a family $\mathcal{V} \subseteq 2^V$ of vertex subsets, $G/\mathcal{V}$ denotes a graph obtained by shrinking each $X \in \mathcal{V}$ into a single vertex. We say that $X \subseteq V$ *separates* $\mathcal{V}$ if either $Y \subseteq X$ or $Y \subseteq V - X$ holds for each $Y \in \mathcal{V}$, and $Y \subseteq X \subseteq V - Y'$ for some $Y, Y' \in \mathcal{V}$.

2.1.3 Digraph

Let $D = (V, A)$ denote a digraph with vertex set V and arc set A, where an arc is defined as an ordered pair of vertices in V. We denote an arc with tail u and head v by uv. In this thesis, D is usually a multi-digraph, i.e., arc set contains some parallel edges, unless stated otherwise. We distinguish two parallel arcs $e_1 = uv$ and $e_2 = uv$, which may be simply denoted by uv and uv.

For two vertex subsets $U, W \subseteq V$, $\delta(U, W; D)$ represents the set of arcs in A whose tail is in U and head is in W. If U consists of only a single vertex v, we represent U by v. Then $\delta(v, v; D)$ is the set of loops in A incident to vertex v. Moreover $\delta(V - U, U; D)$ (resp., $\delta(U, V - U; D)$) is simply denoted by $\delta^-(U; D)$ (resp., $\delta^+(U; D)$). Furthermore we let $d(U, W; D) = |\delta(U, W; D)|$, $d^-(U; D) = |\delta^-(U; D)|$, and $d^+(U; D) = |\delta^+(U; D)|$, respectively. Note that the in-degree $\deg^-(v; D)$ (resp., out-degree $\deg^+(v; D)$) of vertex v is $d(v, v; D) + d^-(v; D)$ (resp., $d(v, v; D) + d^+(v; D)$). A digraph D is called *Eulerian* when $\deg^+(v; D) = \deg^-(v; D)$ for all $v \in V$. When the digraph under consideration is obvious, we may omit to clarify D in these notations.

For a vertex $v \in V$, $\Gamma^-(v)$ (resp., $\Gamma^+(v)$) denotes the set of tails (resp., heads) of arcs entering (resp., leaving) v.

2.1.4 Hypergraph

A hypergraph $H = (V, E)$ with vertex set V and hyperedge set E is a generalization of an undirected graph where each hyperedge is a set of vertices whose size may be larger than two.

For a vertex $v \in V$, we let $\delta(v; H)$ denote the set of hyperedges in E that contain v, i.e., $\delta(v; H) = \{e \in E \mid v \in e\}$. For a hyperedge $e \in E$, $\delta(e; H)$ denotes the set of hyperedges in E that shares at least one vertex with e, i.e., $\delta(e; H) = \cup_{v \in e} \delta(v; H)$. If H is obvious, we may omit to clarify H in these notations.

2.2 Polyhedral aspect of combinatorial optimization

2.2.1 Definitions

Polyhedra have a close relationship to the combinatorial optimization. Actually a large number of memorable advance in the combinatorial optimization was presented by the analysis from the aspect of polyhedra. The purpose of this section is to give several fundamental facts about the combinatorial optimization, linear programming and polyhedra, which will be used in the subsequent chapters.

A *polyhedron* P is defined as $\{x \in \mathbb{R}^n \mid Ax \leq b\}$ with some $A \in \mathbb{R}^{m \times n}$ and $b \in \mathbb{R}^m$. If polyhedron P is represented by a rational matrix $A \in \mathbb{Q}^{m \times n}$ and vector $b \in \mathbb{Q}^m$, then P is called *rational*. A polyhedron is especially called a *polytope* if it is the convex hull of finitely many vectors in $\mathbb{R}^n$. It is known that a polyhedron P is a polytope if and only if P is bounded [70].

A *linear programming* is a problem to minimize (or maximize) an objective function $w^T x$ over a polyhedron P with some cost vector $w \in \mathbb{R}^n$. Instances with cost vector $w = 1^n$ are called *cardinality cases* while the others are called *cost cases*. There are some polynomial time algorithms for a linear programming such as the ellipsoid method and Karmarkar's algorithm. It is known that the running time of the ellipsoid method can be polynomial independent from m if the following separation problem to the feasible region P can be solved in polynomial time independent from m.

Separation problem
Given a polyhedron $\mathrm{P} \subseteq \mathbb{R}^n$ and a vector $x \in \mathbb{Q}^n$, decide whether $x \in \mathrm{P}$ or not. If $x \notin \mathrm{P}$, find a vector $a \in \mathbb{Q}^n$ such that $a^T x < a^T y$ for all $y \in \mathrm{P}$.

In addition, É. Tardos [75, 76] showed that a linear programming admits a strongly polynomial time algorithm when its feasible region is represented by $\{x \in \mathbb{R}^n \mid Ax \leq b, A \in \{0,1\}^{m \times n}, b \in \mathbb{Q}^m\}$.

A polyhedron P is called *integer* if it is the convex hull of the integer vectors contained in P. This is equivalent to that each vertex of P is integer. A linear programming is called *integer* if its feasible region is integer. Hence there always exists an integer optimal solution for an integer linear programming.

A combinatorial optimization can be generally represented by an integer programming, which minimizes (or maximizes) an objective function $w^T x$ over the intersection of a polyhedron P and $\mathbb{Z}^n$. In this case, a linear programming relaxation of the problem is defined as a linear programming over a polyhedron $\mathrm{P}' \supseteq \mathrm{P} \cap \mathbb{Z}^n$. Notice that P can be used as P'. If P' is integer and $\mathrm{P}' - \mathrm{P}$ contains no integer vector, then P' is the convex hull of incidence vectors of all feasible solutions. Hence solving the linear programming relaxation is equivalent to solving the original problem. For problems classified as NP-hard, we cannot expect such a linear programming relaxation unless P = NP.

2.2.2 Connector polytope and spanning tree polyhedron

In this and the subsequent subsections, we present some examples of integer polyhedra defined from combinatorial optimization problems. First, we see polyhedra defined from spanning trees.

A *spanning tree* on a vertex set V is a tree connecting all vertices in V. The *minimum spanning tree problem* is defined as follows.

> **Minimum spanning tree problem**
> Given an undirected graph $G = (V, E)$ and an edge cost $w : E \to \mathbb{Q}$, find a minimum cost spanning tree $F \subseteq E$.

Consider the following integer programming.

$$\begin{array}{lll} \text{minimize} & w^T x & \\ \text{subject to} & x(\delta(U)) \geq 1 & \text{for each non-empty } U \subset V, \\ & x \in \{0,1\}^E. & \end{array} \tag{2.1}$$

Every feasible solution of (2.1) is an incidence vector of $F \subseteq E$ such that (V, F) is connected. We call such F a *connector*. Observe that an inclusion-wise minimal connector is a spanning tree. Hence problem (2.1) is an integer programming formulation of the minimum spanning tree problem when $w(e) \geq 0$ for all $e \in E$.

Unfortunately, the linear programming obtained by relaxing the constraint $x \in \{0,1\}^E$ into $0 \leq x(e) \leq 1$, $e \in E$ is not equivalent to (2.1). For example, let $G = (V, E)$ be a cycle and $w(e) = 1$ for every $e \in E$. Then a solution $x(e) = 1/2$, $e \in E$ achieves the objective value of $|E|/2$. On the other hand, the optimal value of (2.1) is $|E| - 1$ for this instance.

However, we can obtain a linear programming equivalent to the minimum spanning tree with non-negative edge cost. Let CON(G) be the polytope in $\mathbb{R}^E$ determined by the following inequalities.

$$0 \leq x(e) \leq 1 \qquad \text{for each } e \in E, \tag{2.2a}$$

$$x(\delta(\mathcal{P})) \geq |\mathcal{P}| - 1 \qquad \text{for each partition } \mathcal{P} \text{ of } V \text{ into non-empty classes.} \tag{2.2b}$$

It is easy to see that the incidence vector $\mathcal{X}_F \in \{0,1\}^E$ of an edge set $F \subseteq E$ is contained in CON(G) if and only if F is a connector. On the other hand, polytope CON(G) is known to be integer [71]. Hence every extreme point of CON(G) represents the incidence vector of a connector.

Now we let ST(G) be the polyhedron obtained from CON(G) by relaxing (2.2a) into $0 \leq x(e)$, $e \in E$. That is to say, ST(G) is determined by

$$0 \leq x(e) \qquad \text{for each } e \in E, \tag{2.3a}$$

$$x(\delta(\mathcal{P})) \geq |\mathcal{P}| - 1 \qquad \text{for each partition } \mathcal{P} \text{ of } V \text{ into non-empty classes.} \tag{2.3b}$$

Theorem 2.1. *Let $G = (V, E)$ be an undirected graph, and $w : E \to \mathbb{Q}_+$ be a non-negative edge cost. Then $\min\{w^T x \mid x \in ST(G)\}$ is equal to the minimum cost of spanning trees of G.*

Proof. Let $x^* \in \mathbb{R}^E$ be a minimal vector in $\mathrm{ST}(G)$, i.e., no vector $x < x^*$ is contained in $\mathrm{ST}(G)$. In the following, we show that $x^*(e) \leq 1$ holds for every $e \in E$. This implies that $\min\{w^T x \mid x \in \mathrm{ST}(G)\} = \min\{w^T x \mid x \in \mathrm{CON}(G)\}$, which leads to the theorem.

Now we suppose inversely that $x^*(e) > 1$ for some $e \in E$. By the minimality of x^*, there exists a partition $\mathcal{P}$ of V such that $e \in \delta(\mathcal{P})$ and $x^*(\delta(\mathcal{P})) = |\mathcal{P}| - 1$. Clearly $|\mathcal{P}| \geq 3$ holds for this $\mathcal{P}$ since otherwise $x^*(\delta(\mathcal{P})) \geq x^*(e) > 1 \geq |\mathcal{P}| - 1$. Let $U, W \subset V$ be the classes of $\mathcal{P}$ such that $e \in \delta(U)$ and $e \in \delta(W)$, and define $\mathcal{P}'$ as $(\mathcal{P} - \{U, W\}) \cup \{U \cup W\}$. Then $|\mathcal{P}'| = |\mathcal{P}| - 1$, implying that

$$x^*(\delta(\mathcal{P}')) \leq x^*(\delta(\mathcal{P})) - x^*(e) < (|\mathcal{P}| - 1) - 1 = |\mathcal{P}'| - 1.$$

This means that x^* violates (2.3b) for $\mathcal{P}'$, a contradiction. □

2.2.3 Capacitated a-edge cover polytope

Another example is a polytope defined from the *capacitated a-edge cover problem* ((a, c)-edge cover problem). For an undirected graph $G = (V, E)$ together with a demand $a : V \to \mathbb{Z}_+$ of incident edges and a capacity $c : E \to \mathbb{Z}_+$ of multiplicity, a multiple edge set F without loops is called an *(a, c)-edge cover* if $\deg(v; G_F) \geq a(v)$ holds for each $v \in V$ and the number of multiple copies of edge e contained in F is at most $c(e)$ for each $e \in E$. The (a, c)-edge cover problem is defined as follows.

> **(a, c)-edge cover problem**
> Given an undirected graph $G = (V, E)$, a demand $a : V \to \mathbb{Z}_+$ of incident edges, a capacity $c : E \to \mathbb{Z}_+$ of multiplicity, and an edge cost $w : E \to \mathbb{Q}_+$, find a minimum cost (a, c)-edge cover.

The $(a, +\infty)$-edge cover problem is especially called the *a-edge cover problem*, and its feasible solutions are called *a-edge covers*. Furthermore, the 1-edge cover problem is especially called the *edge cover problem*, and its feasible solutions are called *edge covers*.

An integer programming formulation of the (a, c)-edge cover problem is

$$\begin{array}{lll} \text{minimize} & w^T x & \\ \text{subject to} & x(\delta(v)) \geq a(v) & \text{for each } v \in V, \\ & x(e) \leq c(e) & \text{for each } e \in E, \\ & x \in \mathbb{Z}_+^E. & \end{array} \tag{2.4}$$

It is known that there exists a strongly polynomial time algorithm for the (a, c)-edge cover problem [60]. Let $\mathrm{EC}(G, a, c)$ denote the polytope in $\mathbb{R}^E$ determined by the following inequalities.

$$0 \leq x(e) \leq c(e) \qquad \text{for each } e \in E, \tag{2.5a}$$

$$x(\delta(v)) \geq a(v) \qquad \text{for each } v \in V, \tag{2.5b}$$

$$x(E[U]) + x(\delta(U)) - x(F) \geq \left\lceil \frac{a(U) - c(F)}{2} \right\rceil$$
$$\text{for each } U \subseteq V,\ F \subseteq \delta(U) \text{ with odd } a(U) - c(F). \quad (2.5c)$$

All incidence vectors of (a, c)-edge covers are contained in $\mathrm{EC}(G, a, c)$. Hence it gives a linear programming relaxation of the (a, c)-edge cover problem. Furthermore, it is shown [71] that $\mathrm{EC}(G, a, c)$ is an integer polytope. Since integer vectors satisfying conditions (2.5a) and (2.5b) are incidence vectors of (a, c)-edge covers, minimizing $w^T x$ over $\mathrm{EC}(G, a, c)$ gives the minimum cost of (a, c)-edge covers.

If $c = +\infty$ and $F \neq \emptyset$, then (2.5c) is always satisfied because its right hand side equals to $-\infty$. Hence in $\mathrm{EC}(G, a, +\infty)$, (2.5c) can be replaced by

$$x(E[U]) + x(\delta(U)) \geq \lceil a(U)/2 \rceil \quad \text{for each } U \subseteq V \text{ with odd } a(U). \qquad (2.5c')$$

2.2.4 Capacitated a-matching polytope

The last example is defined from the *capacitated a-matching problem* ((a, c)-matching problem). For an undirected graph $G = (V, E)$, let us define two capacities $a : V \to \mathbb{Z}_+$ and $c : E \to \mathbb{Z}_+$. A multiset F of edges without loops is called a *capacitated a-matching* ((a, c)-matching) when $\deg(v; G_F) \leq a(v)$ for each $v \in V$ and the number of copies of $e \in E$ contained in F is at most $c(e)$. Then (a, c)-matching problem is defined as follows.

> **(a, c)-matching problem**
> Given an undirected graph $G = (V, E)$, a capacity $a : V \to \mathbb{Z}_+$ of incident edges, a capacity $c : E \to \mathbb{Z}_+$ of multiplicity, and an edge cost $w : E \to \mathbb{Q}_+$, find a maximum cost (a, c)-matching.

The $(a, +\infty)$-matching problem is especially called the a-matching problem, and its feasible solutions are called a-matchings. Furthermore, the 1-matching problem is especially called the *matching problem*, and its feasible solutions are called *matchings*.

An integer programming formulation of the (a, c)-matching problem is given as follows.

$$\begin{array}{lll} \text{maximize} & w^T x & \\ \text{subject to} & x(\delta(v)) \leq a(v) & \text{for each } v \in V, \\ & x(e) \leq c(e) & \text{for each } e \in E, \\ & x \in \mathbb{Z}_+^E. & \end{array} \qquad (2.6)$$

The (a, c)-matching problem is known to be solvable in strongly polynomial time [1, 71]. In the (a, c)-matching problem, restricting $x(e)$ by $c(e)$ is not essential because all instances of the (a, c)-matching problem can be reduced to those with $c = +\infty$ (see [71] for example).

Let $\mathrm{MA}(G, a, c)$ denote the polytope in $\mathbb{R}^E$ determined by

$$0 \leq x(e) \leq c(e) \qquad \text{for each } e \in E, \qquad (2.7a)$$

$$x(\delta(v)) \leq a(v) \qquad \text{for each } v \in V, \qquad (2.7b)$$

$$x(E[U]) + x(F) \leq \left\lfloor \frac{a(U) + c(F)}{2} \right\rfloor \qquad \begin{array}{l} \text{for each } U \subseteq V,\ F \subseteq \delta(U) \\ \text{with } a(U) + c(F) \text{ odd.} \end{array} \qquad (2.7c)$$

All incidence vectors of (a,c)-matchings are contained in $\mathrm{MA}(G,a,c)$. Hence it gives a linear programming relaxation of the (a,c)-matching problem. Moreover it is shown that $\mathrm{MA}(G,a,c)$ is an integer polytope [73]. Since integer vectors satisfying conditions (2.7a) and (2.7b) are incidence vectors of (a,c)-matchings in G, maximizing $w^T x$ over the polytope $\mathrm{MA}(G,a,c)$ gives the maximum cost of (a,c)-matchings.

On the other hand, a matching is called *perfect* if it covers all vertices (i.e., an edge set F such that $\deg(v;G_F)=1$ for $v \in V$). The *perfect matching problem* is formulated similarly for the (a,c)-matching problem.

Perfect matching problem
Given an undirected graph $G=(V,E)$ and an edge cost $w : E \to \mathbb{Q}$, find a minimum cost perfect matching.

Let $\mathrm{PMA}(G)$ be a polytope determined by

$$x(e) \geq 0 \qquad \text{for each } e \in E, \tag{2.8a}$$
$$x(\delta(v)) = 1 \qquad \text{for each } v \in V, \tag{2.8b}$$
$$x(\delta(U)) \geq 1 \qquad \text{for each } U \subseteq V \text{ with } |U| \text{ is odd.} \tag{2.8c}$$

Similarly for $\mathrm{MA}(G,a,c)$, $\mathrm{PMA}(G)$ is known to be an integer polytope [71]. Since an integer vector is the incidence vector of a perfect matching if and only if it is contained in $\mathrm{PMA}(G)$, minimizing $w^T x$ over $\mathrm{PMA}(G)$ is equivalent to the perfect matching problem.

2.3 Connectivity of graphs

2.3.1 Arc-connectivity, edge-connectivity, and cut function

Let $D=(V,A)$ be a digraph. We say that a non-empty subset X of V or the cut $\delta^+(X;D)$ defined from such X *separates* v from u when $u \in X \subseteq V-v$. The (local) *arc-connectivity* $\lambda(u,v;D)$ from u to v is defined as the minimum size of cuts separating v from u, i.e.,

$$\lambda(u,v;D) = \min\{d^+(X;D) \mid u \in X \subseteq V-v\}.$$

By T. Grünwald (=T. Gallai) [35], who extended the theorem due to K. Menger [58], it is shown that $\lambda(u,v;D)$ equals to the maximum number of arc-disjoint paths from u to v. By this fact, we can see that the transitive law holds for function λ; i.e., for $u,v,z \in V$, $\lambda(u,z;D) \geq k$ follows $\lambda(u,v;D) \geq k$ and $\lambda(v,z;D) \geq k$. By applying a max-flow algorithm (see [51] for example) to digraph D and arc capacity 1^E, we can compute $\lambda(u,v;D)$ in strongly polynomial time.

For a subset $U \subseteq V$ and a function $r : U \times U \to \mathbb{Z}_+$, D is called *r-arc-connected* in U if $\lambda(u,v;D) \geq r(u,v)$ holds for all $u,v \in U$. If $r(u,v)=k$ for every $u,v \in U$ with a positive integer $k \in \mathbb{Z}_+$, then D is called *k-arc-connected* in U. When $U=V$, we simply say that D is r-arc-connected or k-arc-connected. The (global) arc-connectivity $\lambda(D)$ of D is define as

$$\min_{u,v \in V} \lambda(u,v;D) = \min\{d^+(X;D) \mid \emptyset \neq X \subset V\}.$$

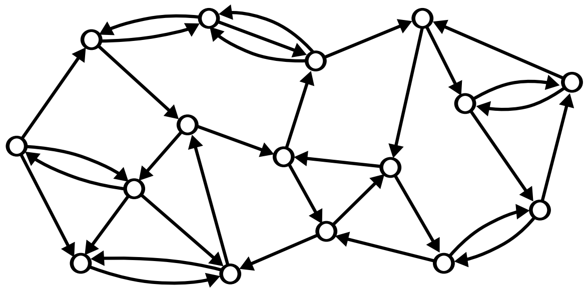


Figure 2.1: An example of 2-arc-connected multi-digraphs

Notice that D is k-arc-connected if $\lambda(D) \geq k$. Figure 2.1 shows an example of multi-digraphs D with $\lambda(D) = 2$. The arc-connectivity of a digraph can be also computed in strongly polynomial time since it suffices to compute the arc-connectivity between all pairs of vertices. Besides, we can see that $|V| - 1$ times of computing the max-flow are enough from the transitive law of the arc-connectivity.

A *k-arc-connected component* is defined as an inclusion-wise maximal subset X of V such that $\lambda(u, v; D) \geq k$ holds for $u, v \in X$, where k is possibly larger than $\lambda(D)$.

The following presents a useful property of cuts.

Theorem 2.2. *For any $X, Y \subset V$, it holds*

$$d^+(X; D) + d^+(Y; D) = d^+(X \cap Y; D) + d^+(X \cup Y; D) \\ + d(X - Y, Y - X; D) + d(Y - X, X - Y; D), \quad (2.9)$$

and

$$d^-(X; D) + d^-(Y; D) = d^-(X \cap Y; D) + d^-(X \cup Y; D) \\ + d(X - Y, Y - X; D) + d(Y - X, X - Y; D). \quad (2.10)$$

Proof. They can be proven by counting both sides. □

If cut sizes are symmetric, the following property also holds.

Theorem 2.3. *Suppose $d^+(X) = d^-(X)$ for every $X \subset V$. Then*

$$d^+(X; D) + d^+(Y; D) = d^+(X - Y; D) + d^+(Y - X; D) \\ + d(X \cap Y, V - (X \cup Y); D) + d(V - (X \cup Y), X \cap Y; D) \quad (2.11)$$

holds for any $X, Y \subset V$.

Proof. Since $d^+(Y) = d^-(V - Y) = d^+(V - Y)$, it holds

$$d^+(X; D) + d^+(Y; D) = d^+(X; D) + d^+(V - Y; D). \quad (2.12)$$

From (2.9) for X and $V - Y$, we have

$$d^+(X; D) + d^+(V - Y; D) = d^+(X \cap (V - Y); D) + d^+(X \cup (V - Y); D) \\ + d(X - (V - Y), (V - Y) - X; D) + d((V - Y) - X, X - (V - Y); D).$$

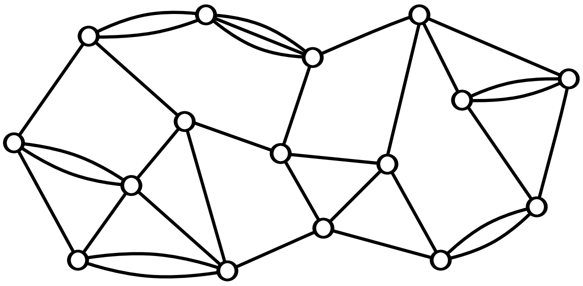


Figure 2.2: An example of 4-edge-connected multigraphs

Observe that $X \cap (V-Y) = X - Y$, $X \cup (V-Y) = V - (Y - X)$, $X - (V-Y) = X \cap Y$ and $(V-Y) - X = V - (X \cup Y)$ hold. Therefore the right hand side of (2.12) equals to

$$d^+(X-Y) + d^+(Y-X) + d^+(X \cap Y, V - (X \cup Y)) + d^+(V - (X \cup Y), X \cap Y),$$

as required. □

Let $G = (V, E)$ be an undirected graph. We say that a non-empty subset X of V or the cut $\delta(X)$ defined from such X *separates* two vertices when X contains exactly one of them. The (local) *edge-connectivity* $\lambda(u, v; G)$ between two vertices u and v is the minimum size of cuts separating u and v, i.e.,

$$\lambda(u, v; G) = \min\{d(X; G) \mid u \in X \subseteq V - v\}.$$

In contrast to digraphs, it always holds that $\lambda(u, v; G) = \lambda(v, u; G)$ here. Let D_G be the digraph obtained from G by replacing each edge uv with two arcs uv and vu. Then $d(X; G) = d^+(X; D_G)$ holds, which implies that $\lambda(u, v; G) = \lambda(u, v; D_G)$. By the relationship between the arc-connectivity and the number of arc-disjoint paths in D_G, we can see that $\lambda(u, v; G)$ is equivalent to the maximum number of edge-disjoint paths joining u and v. Moreover, the transitive law also holds for the edge-connectivity; i.e., for $u, v, z \in V$, $\lambda(u, z; G) \geq k$ follows $\lambda(u, v; G) \geq k$ and $\lambda(v, z; G) \geq k$.

For a subset $U \subseteq V$ and a function $r : \binom{U}{2} \to \mathbb{Z}_+$, G is called *r-edge-connected* in U if $\lambda(u, v; G) \geq r(u, v)$ holds for all $u, v \in U$. If $r(u, v) \geq k$ for every $u, v \in U$ with a positive integer $k \in \mathbb{Z}_+$, G is called *k-edge-connected* in U. When $U = V$, we simply say that G is r-edge-connected or k-edge-connected. The (global) edge-connectivity $\lambda(G)$ of G is defined as

$$\min_{u,v \in V} \lambda(u, v; G) = \min\{d(X; G) \mid \emptyset \neq X \subset V\}.$$

Notice that G is k-edge-connected if $\lambda(G) \geq k$. Figure 2.2 shows an example of multigraphs G with $\lambda(G) = 4$.

A *k-edge-connected component* is defined as an inclusion-wise maximal subset X of V such that $\lambda(u, v; G) \geq k$ holds for $u, v \in X$, where k is possibly larger than $\lambda(G)$. The strong polynomiality of computing the edge-connectivity also follows from that of the arc-connectivity.

Theorem 2.4. *For any $X, Y \subset V$, it holds*

$$d(X;G) + d(Y;G) = d(X \cap Y;G) + d(X \cup Y;G) + 2d(X - Y, Y - X;G) \tag{2.13}$$

and

$$d(X;G) + d(Y;G) = d(X - Y;G) + d(Y - X;G) + 2d(X \cap Y, V - (X \cup Y);G). \tag{2.14}$$

Proof. Inequality (2.13) is immediate from (2.9) (or (2.10)) in Theorem 2.2 for D_G. Inequality (2.14) is immediate from (2.13) in Theorem 2.3 for D_G. $\square$

Since cut sizes are non-negative, we can derive

$$d(X;G) + d(Y;G) \geq d(X \cap Y;G) + d(X \cup Y;G) \tag{2.15}$$

for any $X, Y \subset V$ from (2.13). This property of a set function is called *submodularity*.

In this thesis, we sometimes discuss the edge-connectivity in an undirected graph $G = (V, E)$ whose edge set is weighted by a vector $x \in \mathbb{R}_+^E$. In this case, we assume without loss of generality that G is the complete graph on V by augmenting E with edges $e \in \binom{V}{2} - E$, where we let $x(e) = 0$, $e \in \binom{V}{2} - E$. We denote the set of G and x by (V, x). In (V, x), the edge-connectivity $\lambda(u, v; V, x)$ between two vertices u and v is defined as

$$\min\{x(\delta(X)) \mid u \in X \subset V - v\}.$$

Other notions such as r-edge-connectivity and k-edge-connected components in (V, x) are defined from $\lambda(u, v; V, x)$ analogously to the ordinary graphs.

2.3.2 Steiner network problem

One of the most basic problems in the network design is to construct a minimum cost r-edge-connected graph, which is called the *Steiner network problem* or the *survivable network problem*.

> **Steiner network problem (Survivable network problem)**
> Given a vertex set V, an edge cost $w : \binom{V}{2} \to \mathbb{Q}_+$, an edge capacity $c : \binom{V}{2} \to \mathbb{Z}_+$, and an edge-connectivity requirement $r : \binom{V}{2} \to \mathbb{Z}_+$, find a minimum cost edge set E such that (V, E) is r-edge-connected graph and E contains at most $c(uv)$ edges between u and v.

We especially call the problem of instances with $c = +\infty$ the Steiner network problem *without edge capacity*. An edge cost w is called *metric* if it obeys the triangle inequality. In other words,

$$w(uv) + w(vz) \geq w(uz)$$

holds for any three vertices u, v and z. For the Steiner network problem without edge capacity, it preserves the generality to assume that the edge cost is metric because an instance with any edge cost w can be reduced to one with a metric edge cost w', where

$$w'(uv) = \min\{w(P) \mid P \text{ is a path whose both ends are } u \text{ and } v\}.$$

A linear programming relaxation of the Steiner network problem without edge capacity is given as

$$\begin{array}{ll} \text{minimize} & w^T x \\ \text{subject to} & x(\delta(X)) \geq \max_{u \in X, v \in V-X} r(u,v) \quad \text{for each } X \subset V,\ X \neq \emptyset, \\ & x \in \mathbb{R}_+^{\binom{V}{2}}. \end{array} \tag{2.16}$$

The *parsimonious property* of the Steiner network problem due to M. X. Goemans and D. J. Bertsimas [30] tells that if w is a metric cost, then the above linear programming is equivalent to

$$\begin{array}{lll} \text{minimize} & w^T x & \\ \text{subject to} & x(\delta(X)) \geq \max_{u \in X, v \in V-X} r(u,v) & \text{for each } X \subset V,\ X \neq \emptyset, \\ & x(\delta(v)) = \max_{u \in V-v} r(u,v) & \text{for each } v \in V, \\ & x \in \mathbb{R}_+^{\binom{V}{2}}. & \end{array}$$

This property is proven by showing that there always exists an optimal solution x^* of (2.16) which satisfies $x^*(\delta(v)) = \max_{u \in V-v} r(u,v)$ in that case. In Chapter 4, we extend this property to an integer programming.

The Steiner network problem is known to be NP-hard [77]. With regards to the algorithmic results, a 2-approximation algorithm is proposed by K. Jain [45]. Actually, his algorithm can be applicable to more general problem than the Steiner network problem. We will describe this in the following.

A set function $f : 2^V \to \mathbb{Z}_+$ with $f(V) = f(\emptyset) = 0$ is called *weakly supermodular* (or *skew supermodular*) if, for every $X, Y \subseteq V$,

$$f(X) + f(Y) \leq f(X \cap Y) + f(X \cup Y) \tag{2.17}$$

or

$$f(X) + f(Y) \leq f(X - Y) + f(Y - X) \tag{2.18}$$

holds. The following theorem gives an example of weakly supermodular functions.

Theorem 2.5. *Let $r : \binom{V}{2} \to \mathbb{Z}_+$, and define a set function*

$$f_r(X) = \begin{cases} 0 & \text{if } X = V \text{ or } X = \emptyset, \\ \max\{r(u,v) \mid u \in X, v \notin X\} & \text{otherwise} \end{cases} \tag{2.19}$$

from r. Then f_r is weakly supermodular. If $X \cup Y = V$, then (2.18) *always holds for f_r.*

Proof. First observe that replacing Y by $V - Y$ transforms each of (2.17) and (2.18) into the other. Hence if we show one of those inequalities holds for X and $V - Y$, the other inequality holds for X and Y. This means that we can always replace Y by $V - Y$ if necessary.

We suppose without loss of generality that $f_r(X) \geq f_r(Y)$. Let vertices $u^* \in X$ and $v^* \in V - X$ satisfy $f_r(X) = r(u^*, v^*)$. By replacing Y by $V - Y$ if necessary, we can let $u^* \notin Y$. Now we let $v^* \notin Y$ firstly. Then $f_r(X \cup Y) = f_r(X - Y) = r(u^*, v^*)$ holds. Since $f_r(Y) \leq f_r(X \cap Y)$ or $f_r(Y) \leq f_r(Y - X)$ always holds, (2.17) or (2.18) follows in this case. Secondly let $v^* \in Y$. Then $f_r(X) = f_r(Y) = f_r(X - Y) = f_r(Y - X) = r(u^*, v^*)$ holds. Accordingly (2.18) holds. Moreover notice that, if $X \cup Y$, then $v^* \in Y$, and hence (2.18) holds. These facts complete the proof. $\square$

2.4 Approximation algorithm

2.4.1 Definitions

Let us consider a minimization problem with a set $\mathcal{I}$ of instances. For an instance $I \in \mathcal{I}$, let $\mathrm{OPT}(I)$ denote its optimal cost. An algorithm is called an *α-approximation algorithm* when it outputs a feasible solution of the cost at most $\alpha\mathrm{OPT}(I)$ for all $I \in \mathcal{I}$. In this case, α is called an *approximation factor*.

For analyzing the approximation factor of an algorithm to a combinatorial optimization problem, its linear programming relaxation often plays an important role. In the analysis, the cost of a solution output by the algorithm is estimated in terms of a lower bound on the optimal cost. As this lower bound, the optimal cost of a linear programming relaxation is often used.

Let $\mathrm{LP}(I)$ be a linear programming relaxation for an instance $I \in \mathcal{I}$, and $\mathrm{OPT}(\mathrm{LP}(I))$ denote the optimal cost of $\mathrm{LP}(I)$. The *integrality gap* of the linear programming relaxation is defined as

$$\sup_{I \in \mathcal{I}} \frac{\mathrm{OPT}(I)}{\mathrm{OPT}(\mathrm{LP}(I))}.$$

The integrality gap gives an upper bound on the approximation factor obtained by the analysis which uses $\mathrm{OPT}(\mathrm{LP}(I))$ as a lower bound on $\mathrm{OPT}(I)$. Therefore, a relaxation of small integrality gap is favorable for obtaining the better approximation factor.

For a maximization problem, an α-approximation algorithm is defined as an algorithm which outputs a feasible solution of the cost at least $\alpha\mathrm{OPT}(I)$ for all $I \in \mathcal{I}$. The integrality gap of the linear programming relaxation is defined as

$$\inf_{I \in \mathcal{I}} \frac{\mathrm{OPT}(I)}{\mathrm{OPT}(\mathrm{LP}(I))}.$$

From now on, we review several basic approximation algorithms.

2.4.2 Approximation algorithm to the set cover problem

The *set cover problem* is described as follows.

> **Set cover problem**
> Given a set V of elements, a family $\mathcal{S} \subseteq 2^V$, and a cost $w : \mathcal{S} \to \mathbb{Q}_+$, find a minimum cost family $\mathcal{T} \subseteq \mathcal{S}$ such that $\cup_{T \in \mathcal{T}} T = V$.

A feasible solution for the set cover problem is called a *set cover*. This problem is proven to be NP-hard [28]. We sometimes regard $(V, \mathcal{S})$ as a hypergraph. Notice that the set cover problem can be regarded as a hypergraph version of the edge cover problem. The set cover problem can be approximated by the following greedy algorithm [13, 46, 54].

Algorithm SETCOVER

Input: A set V of elements, a family $\mathcal{S} \subseteq 2^V$, and a cost $w : \mathcal{S} \to \mathbb{Q}_+$

Output: A set cover $\mathcal{T} \subseteq \mathcal{S}$

$\mathcal{T} := \emptyset$;
while $V \neq \cup_{T \in \mathcal{T}} T$ **do**
  $S :=$ a set in $\mathcal{S} - \mathcal{T}$ minimizing $w(S)/|S - \cup_{T\in\mathcal{T}} T|$;
  $\mathcal{T} := \mathcal{T} \cup S$
end while;
Output $\mathcal{T}$;

The following is an integer programming formulation of the set cover problem.

$$\begin{array}{lll} \text{minimize} & w^T x & \\ \text{subject to} & x(\delta(v)) \geq 1 & \text{for each } v \in V, \\ & x \in \{0,1\}^{\mathcal{S}}. & \end{array}$$

Let $\mathrm{SC}(V, \mathcal{S})$ denote the feasible region of the linear programming obtained from the formulation by relaxing the constraint $x \in \{0,1\}^{\mathcal{S}}$ into $x \geq 0^{\mathcal{S}}$. Moreover, define OPT_{SC} as the optimal cost of the linear programming, i.e., $\min\{w^T x \mid x \in \mathrm{SC}(V,\mathcal{S})\}$. By the duality of the linear programming (see [70] for example), OPT_{SC} is also the optimal value of the following linear programming.

$$\begin{array}{lll} \text{maximize} & \sum_{v\in V} y(v) & \\ \text{subject to} & \sum_{v \in S} y(v) \leq w(S) & \text{for each } S \in \mathcal{S}, \\ & y \in \mathbb{R}_+^V. & \end{array} \tag{2.20}$$

The cost of a solution output by algorithm SETCOVER can be estimated in terms of OPT_{SC} and the maximum size of sets.

Theorem 2.6 ([13, 54]). *Algorithm SETCOVER outputs a solution of cost at most $\theta_k \mathrm{OPT}_{SC}$ to the set cover problem, where $k = \max_{S\in\mathcal{S}} |S|$.*

Proof. For an element $v \in V$, let $S_v \in \mathcal{S}$ denote the set covering v (i.e., $v \in S$) which is selected by algorithm SETCOVER earliest as a member of the solution. We then define the price $p(v)$ of v as $w(S_v)/|S_v - \cup_{T\in\mathcal{T}} T|$, where $\mathcal{T}$ is the solution maintained by the algorithm when S_v is selected. Moreover, let $y \in \mathbb{R}_+^V$ be the vector such that $y(v) = p(v)/\theta_k$. In what follows, we show that y is a feasible solution to the linear programming (2.20). Since algorithm SETCOVER outputs a solution of cost at most $\sum_{v\in V} p(v)$, this proves the theorem.

Let $S = \{v_1, \ldots, v_{|S|}\} \in \mathcal{S}$ be a set selected by the algorithm as a member of the solution output. Without loss of generality, we assume that $v_1, \ldots, v_{|S|}$ represents the order in which they are covered by the algorithm, where ties are arbitrarily broken. When S_{v_i} is selected, S can cover v_i with the average cost of $w(S)/(|S| - i + 1)$. Hence $p(v_i) \leq w(S)/(|S| - i + 1)$. Accordingly, it holds

$$\sum_{i=1}^{|S|} y(v_i) = \sum_{i=1}^{|S|} \frac{p(v_i)}{\theta_k} \leq \frac{w(S)\theta_{|S|}}{\theta_k} \leq w(S),$$

which implies that y is feasible to (2.20). $\square$

This theorem indicates that the approximation factor of algorithm SETCOVER is at most θ_k since OPT_{SC} is a lower bound on the cost of optimal solutions.

Corollary 2.1. *Algorithm SETCOVER is a θ_k-approximation algorithm to the set cover problem, where $k = \max_{S \in \mathcal{S}} |S|$.*

We can also see a lower bound on the approximation factor as follows.

Theorem 2.7 ([17])**.** *The set cover problem admits no $(1-\epsilon)\ln|V|$-approximation algorithm with any $\epsilon > 0$ unless $\mathrm{NP} \subset \mathrm{DTIME}(|V|^{\log\log|V|})$.* □

2.4.3 Traveling salesperson problem

Next, we introduce two approximation algorithms for the TSP introduced in Section 1.1. In general, the TSP cannot be approximated at all as described in the following.

Theorem 2.8 ([68])**.** *For any polynomial time computable function $\alpha(|V|)$, TSP admits no $\alpha(|V|)$-approximation algorithm unless $\mathrm{P} = \mathrm{NP}$.* □

Because of this hardness, the TSP is often discussed with the metric edge cost. We call TSP with metric edge cost *metric TSP*. Now let us see a 2-approximation algorithm for the metric TSP. This algorithm uses the well-known theorem due to L. Euler that every Eulerian graph (i.e., the degree of each vertex is even) has an Eulerian tour (i.e., a tour that traverses all edges exactly once).

Algorithm METRICTSP
Input: A vertex set V and a metric edge cost $w : \binom{V}{2} \to \mathbb{Q}_+$
Output: A Hamiltonian cycle H

1: $T :=$ minimum cost tree spanning V;
2: $H :=$ duplicate of T;
3: **while** $d(v; G_H) > 2$ for some $v \in V$ **do**
4: $\quad xv, vy :=$ edges incident to v in H which appear consecutively in an Eulerian tour of G_H;
5: $\quad H := (H - \{xv, vy\}) \cup \{xy\}$
6: **end while;**
7: Output H;

Theorem 2.9. *Algorithm METRICTSP is a 2-approximation algorithm for the metric TSP.*

Proof. Since T is a spanning tree and every Hamiltonian cycle on V connects all vertices, it holds that $w(T) \leq \mathrm{OPT}$. Let $2T$ denote the duplicate of the minimum cost spanning tree T. We can see that $2T$ is a connected Eulerian graph and $w(2T) = 2w(T) \leq 2\mathrm{OPT}$, where OPT is the minimum cost of Hamiltonian cycles.

Now let us consider the operation that replaces $\{xv, vy\}$ by $\{xy\}$ in Step 5 of algorithm METRICTSP. Firstly, this preserves the property of the graph that is Eulerian since it decreases the degree of v by 2 while maintaining the degrees of the other vertices. Hence repeating this operation transforms the graph into a Hamiltonian cycle. Secondly, this operation preserves the 2-edge-connectivity of the graph because it maintains an Eulerian tour by the way of choosing edges xv and vy. Finally, this operation does not increase the cost of graphs since $w(xv) + w(vy) \geq w(xy)$ holds.

By the first and the second observations, we can see that algorithm METRICTSP outputs a feasible solution H. By the last observation, we can see that $w(H) \leq w(2T)$. Since $w(2T) \leq 2\mathrm{OPT}$, it follows that $w(H) \leq 2\mathrm{OPT}$. Therefore, algorithm METRICTSP is a 2-approximation algorithm. $\square$

Approximation factor of METRICTSP was improved by N. Christofides. His algorithm uses another Eulerian graph of smaller cost instead of the duplicate of a minimum spanning tree.

Algorithm CHRISTOFIDES

Input: A vertex set V and a metric edge cost $w : \binom{V}{2} \to \mathbb{Q}_+$

Output: A Hamiltonian cycle H

$T :=$ minimum cost tree spanning V;
$M :=$ minimum cost perfect matching on the set of vertices whose degrees are odd in G_T;
$H := T \cup M$;
while $d(v; G_H) > 2$ for some $v \in V$ **do**
  $xv, vy :=$ two edges incident to v in H which appear consecutively in an Eulerian tour of G_H;
  $H := (H - \{xv, vy\}) \cup \{xy\}$
end while;
Output H;

Let OPT_{TSP} be the optimal value of

$$\begin{array}{lll} \text{minimize} & w^T x & \\ \text{subject to} & x(\delta(U)) \geq 2 & \text{for each non-empty } U \subset V, \\ & x \in \mathbb{R}^E_+. & \end{array} \tag{2.21}$$

Since (2.21) is a linear programming relaxation of the TSP, OPT_{TSP} is at most the optimal value of the TSP. We can see that the cost of a solution output by algorithm CHRISTOFIDES is at most $1.5\mathrm{OPT}_{TSP}$, which implies that the approximation factor of the algorithm is 1.5. In order to prove this fact, we need the following two lemmas.

Lemma 2.1. *Let T be a minimum spanning tree on V. Then $w(T) \leq \mathrm{OPT}_{TSP}$ holds if edge cost w is metric.*

Proof. Observe that a metric edge cost is non-negative. Hence Theorem 2.1 tells that $w(T) = \min\{w^T x \mid x \in \mathrm{ST}(G)\}$. Therefore the lemma is proven by showing that inequalities (2.3a) and (2.3b) can be derived from the constraints of (2.21). Inequality (2.3a) is obvious. By summing $x(\delta(U)) \geq 2$ over all $U \in \mathcal{P}$, we obtain $x(\delta(\mathcal{P})) \geq |\mathcal{P}|$. Since $|\mathcal{P}| > |\mathcal{P}| - 1$, it implies (2.3b). $\square$

Lemma 2.2. *Let $X \subseteq V$ be a vertex set of even cardinality, and M be a perfect matching on X minimizing $w(M)$. Then $w(M) \leq \mathrm{OPT}_{TSP}/2$ holds if edge cost w is metric.*

Proof. Let x^* be an optimal solution of

$$\begin{array}{rll} \text{minimize} & w^T x & \\ \text{subject to} & x(\delta(U)) \geq 2 & \text{for each non-empty } U \subset X, \\ & x \in \mathbb{R}^E_+. & \end{array}$$

Since this linear programming is a relaxation of (2.21), $w^T x^* \leq \mathrm{OPT}_{TSP}$ holds. By applying the parsimonious property of the Steiner network problem (see Section 2.3.2), we can assume without loss of generality that $x^*(\delta(v)) = 2$ for all $v \in X$ and $x^*(\delta(v)) = 0$ for all $v \in V - X$ (hence $x^*(e) = 0$ for $e \notin E[X]$). Then we can see that $x^*_{E[X]}/2 \in \mathrm{PMA}(G[X])$. Since $\mathcal{X}_M$ achieves the minimum cost over $\mathrm{PMA}(G[X])$, it is indicated that

$$w(M) \leq \frac{w^T x^*_{E[X]}}{2} = \frac{w^T x^*}{2} \leq \frac{\mathrm{OPT}_{TSP}}{2}.$$

□

Then we can obtain the following theorem.

Theorem 2.10 ([30, 78]). *Algorithm CHRISTOFIDES outputs a solution of cost at most* $1.5\mathrm{OPT}_{TSP}$ *to the metric TSP.*

Proof. Algorithm CHRISTOFIDES outputs a solution of cost at most $w(T)+w(M)$, where T is a minimum cost spanning tree on V and M is a minimum cost perfect matching on vertices of odd degrees in G_T. Hence the theorem is immediate from Lemmas 2.1 and 2.2. □

Corollary 2.2. *Algorithm CHRISTOFIDES is a* 1.5*-approximation algorithm for the metric TSP.*

Proof. Immediate from Theorem 2.10 since OPT_{TSP} is at most the cost of optimal solutions. □

A problem of finding a minimum cost directed Hamiltonian cycle can be considered as the digraph version of the TSP. If an arc cost w is metric and symmetric (i.e., $w(uv) = w(vu)$ for every $u, v \in V$), the problem can be approximated within 2 and 1.5 as in the same ways with algorithms METRICTSP and CHRISTOFIDES. The problem with an asymmetric metric arc cost is known to be a hard problem to approximate. The best approximation factor to this is $4/3 \log_3 |V|$ due to H. Kaplan et al. [48]. It seems hard to approximate within a constant factor although the hardness results obtained so far are considerably weaker than this expectation. The best hardness result now is due to C. H. Papadimitriou and S. Vempala [64] telling that it is NP-hard to approximate within a factor less than 117/116.

2.4.4 Generalized Steiner network problem

Here we consider a generalization of the Steiner network problem.

Generalized Steiner network problem
Given a vertex set V, an edge cost $w : \binom{V}{2} \to \mathbb{Q}_+$, an edge capacity $c : \binom{V}{2} \to \mathbb{Z}_+$ and a demand function $f : 2^V \to \mathbb{Z}_+$, find a minimum cost edge set E such that $d(X) \geq f(X)$ holds for every non-empty $X \subset V$.

Observe that this problem is equivalent to the Steiner network problem if function f_r defined in (2.19) is used as a demand function. Analogously to the Steiner network problem, we call the problem consisting of instances with $c = +\infty$ the generalized Steiner network problem *without edge capacity.* The following is the result given by K. Jain [45].

Theorem 2.11 ([45]). *The generalized Steiner network problem can be approximated within a factor of* 2 *if the demand function f is weakly supermodular.* □

Note that the Jain's algorithm first solves a linear programming relaxation of the generalized Steiner network problem, and then rounds the obtained solution to an approximate solution by using the technique called iterated rounding. Since we have already seen that f_r is weakly supermodular in Theorem 2.5, Theorem 2.11 indicates that the Steiner network problem can be approximated within 2.

2.5 Graph transformation: splitting and detachment

The *splitting* is a kind of graph transformation techniques and the *detachment* is its generalization. They are so important by both the theoretical and the practical reasons. In fact, we derive our new results on those techniques in Chapter 4, and utilize them to develop algorithms to some network design problem in Chapter 5. The purpose of this section is to introduce fundamental results on them.

2.5.1 Splitting in graphs

For an undirected graph $G = (V, E)$ and a vertex $s \in V$, *splitting* a pair $\{e = us, f = vs\}$ of edges incident to s indicates an operation that replaces e and f by a new edge uv. We note that e and f are possibly loops incident to s. We let $G^{e,f}$ denote the graph obtained by splitting $\{e, f\}$. The edge-connectivity in $G^{e,f}$ is equal to or smaller than that in G. For example, see Figure 2.3, which illustrates that the edge-connectivity $\lambda(G)$ is decreased from 2 to 1 by splitting $\{e, f\}$. The pair $\{e, f\}$ is called *splittable* if $\lambda(x, y; G^{e,f}) = \lambda(x, y; G)$ holds for all $x, y \in V - s$.

The following theorem due to W. Mader [57] characterizes a condition for graphs to have splittable pairs, answering an earlier conjecture by L. Lovász.

Theorem 2.12 ([57]). *Let $G = (V, E)$ be an undirected connected graph and $s \in V$ be a vertex with* $\deg(s; G) \neq 3$. *If no cut-edge is incident to s, then there exists at least one splittable pair of edges incident to s.* □

A simple proof of this theorem was proposed by A. Frank in [22]. In the same article, he gave the following variant of Theorem 2.12.

Theorem 2.13 ([22]). *Let $G = (V, E)$ be an undirected graph and s be a vertex in V with no incident loop. If no cut-edge is incident to s and* $\deg(s; G)$ *is even, then edges incident to s can be partitioned into* $\deg(s; G)/2$ *disjoint splittable pairs.* □

Frank's proof of Theorem 2.13 is based on the following characterization of splittable pairs.

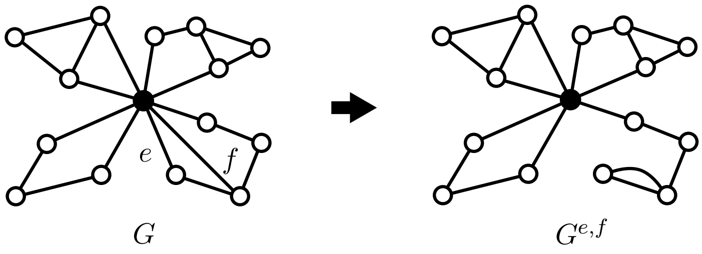


Figure 2.3: Splitting in an undirected graph

Lemma 2.3 ([22]). *Let $f_\lambda(X) = \max_{x \in X, y \in V-(X \cup s)} \lambda(x, y; G)$ for an undirected graph $G = (V, E)$ with a specified vertex s and $X \subset V - s$. A pair $\{e = us, f = vs\}$ of edges is splittable if and only if $f_\lambda(X) - d(X; G) \leq 1$ does not hold for all set $X \subset V - s$ with $\{u, v\} \subseteq X$.* □

The above theorems have been used as a useful technique for solving many connectivity problem. In particular, they play a key role in solving the edge-connectivity augmentation problem (see [21]).

For digraphs, the splittability of a pair of two arcs, one leaving s and the other entering s is defined analogously to undirected graphs. A counterpart of Theorem 2.12 in Eulerian digraphs was proven by A. Frank [20] and B. Jackson [44]

2.5.2 Splitting in edge-weighted graphs

In Chapter 6, we use the splitting in an edge-weighted undirected graph (V, x). In this situation, *splitting* a pair $\{sa, sb\}$ of edges by $\epsilon > 0$ is an operation that decreases $x(sa)$ and $x(sb)$ by ϵ and increases $x(ab)$ by ϵ, where possibly $a = b$ and ϵ is supposed to be at most $\min\{x(su), x(sv)\}$. We execute the splitting in order to isolate a vertex $s \in V$, i.e., $x(sv) = 0$ for every $v \in V - s$. A *complete splitting at* s denotes an operation that isolates s by repeating splitting edges incident to s. The following theorem tells that it always can be executed in strongly polynomial time while preserving the edge-connectivity between every two vertices in $V - s$.

Theorem 2.14. *Let s be an arbitrary vertex in (V, x). There exists a complete splitting at s such that $\lambda(u, v; V, x) = \lambda(u, v; V - s, x')$ holds for every $u, v \in V - s$, where $x' \in \mathbb{R}^{\binom{V-s}{2}}$ is the resulting edge weight from the complete splitting. Such a complete splitting can be found in strongly polynomial time.* □

In the rest of this subsection, we give a proof of this theorem. First, we define $\epsilon_{a,b} \in \mathbb{R}_+$ as the maximum value such that splitting $\{sa, sb\}$ by $\epsilon_{a,b}$ preserves the edge-connectivity between every two vertices in $V - s$. Notice that $\epsilon_{a,b} = \min\{x(sa), x(sb), q_x(a, b)\}$, where

$$q_x(a, b) = \frac{1}{2} \min\{x(\delta(X)) - \lambda(u, v; V, x) \mid a, b, u \in X \subseteq V - (s \cup v), s \neq v\}, \qquad (2.22)$$

because splitting $\{sa, sb\}$ by ϵ decreases $x(\delta(X))$ by 2ϵ if $a, b \in X \subset V - s$, and does not change $x(\delta(X))$ otherwise.

For each $X \subset V - s$, we define

$$f_\lambda(X) = \max\{\lambda(u, v; V, x) \mid u \in X, v \in V - (X \cup s)\},$$

and

$$h(X) = x(\delta(X)) - f_\lambda(X).$$

Notice that $q_x(a,b) = \frac{1}{2}\min_{X:a,b\in X\subset V-s} h(X)$. By Menger's theorem, $h(X) \geq 0$ for every $X \subset V - s$. The following property of h can be derived.

Lemma 2.4. *For every $X, Y \subseteq V - s$, it always holds that*

$$h(X) + h(Y) \geq h(X \cap Y) + h(X \cup Y), \tag{2.23}$$

or

$$h(X) + h(Y) \geq h(X - Y) + h(Y - X) + 2x(\delta(X \cap Y, V - X \cup Y)). \tag{2.24}$$

If $X \cup Y = V - s$, then (2.24) *holds.*

Proof. As in Theorem 2.4, we can see that

$$x(\delta(X)) + x(\delta(Y)) \geq x(\delta(X \cap Y)) + x(\delta(X \cup Y)) \tag{2.25}$$

and

$$x(\delta(X)) + x(\delta(Y)) = x(\delta(X - Y)) + x(\delta(Y - X)) + 2x(\delta(X \cap Y, V - X \cup Y)) \tag{2.26}$$

hold. On the other hand, f_λ is weakly supermodular by Theorem 2.5. In other words, it holds that

$$f_\lambda(X) + f_\lambda(Y) \leq f_\lambda(X \cap Y) + f_\ell(X \cup Y) \tag{2.27}$$

or

$$f_\lambda(X) + f_\lambda(Y) \leq f_\lambda(X - Y) + f_\lambda(Y - X). \tag{2.28}$$

If (2.27) holds, we obtain (2.23) by subtracting (2.27) from (2.25). If (2.28) holds, we obtain (2.24) by subtracting (2.28) from (2.26). As stated in Theore 2.5, (2.28) holds if $X \cup Y = V - s$. Hence (2.24) holds in this case. $\square$

Then we can obtain a fractional version of Theorem 2.13.

Lemma 2.5. *Let sa be an edge such that $x(\delta(s)) > x(sa) > 0$. Then there always exists another edge sb with $\epsilon_{a,b} > 0$.*

Proof. Suppose conversely that $\epsilon_{a,b} = 0$ holds for all $b \in V - (s \cup a)$. Let $N_x(s) = \{b \in V - (s \cup a) \mid x(sb) > 0\}$, where $N_x(s) \neq \emptyset$ holds by the assumption. For every $b \in N_x(s)$, $\epsilon_{a,b} = 0$ indicates $q_x(a,b) = 0$, which implies the existence of $X_b \subset V - s$ such that $a, b \in X_b$ and $h(X_b) = 0$.

Now consider the case of $|N_x(s)| > 2$. Let b_1 and b_2 be two vertices in $N_x(s)$. Since $a \in X_{b_1} \cap X_{b_2}$, $x(\delta(X_{b_1} \cap X_{b_2}, V - X_{b_1} \cup X_{b_2})) \geq x(sa) > 0$ holds. This means that (2.24) does not holds for X_{b_1} and X_{b_2} since the left hand side of (2.24) is $h(X_{b_1}) + h(X_{b_2}) = 0$. Hence (2.23) holds for X_{b_1} and X_{b_2}, and $X_{b_1} \cup X_{b_2} \neq V - s$ by Lemma 2.4. Notice that (2.23) indicates $h(X_{b_1} \cup X_{b_2}) \leq h(X_{b_1}) + h(X_{b_2}) = 0$.

By repeating applying this observation for vertices in $N_x(s)$, we can see that there exists a set $X^* \subset V - s$ with $a \cup N_x(s) \subseteq X^*$ and $h(X^*) = 0$. Define $u \in X^*$ and $v \in V - (s \cup X^*)$ as two vertices with $x(\delta(X^*)) = \lambda(u, v; V, x)$. However, $x(\delta(X^*)) > x(\delta(X^* \cup s)) \geq \lambda(u, v; V, x)$ holds because $N_x(s) \subseteq X^*$, $sa \in \delta(X^*)$ and $x(sa) > 0$. Accordingly, we have a contradiction. $\square$

If s has only one incident edge sv with $x(sv) > 0$, it suffices for isolating s to set $x'(sv) = 0$, which can be regarded as splitting a pair of two edges between s and v by $x(sv)/2$. Otherwise, splitting a pair $\{sa, sb\}$ by $\epsilon_{a,b} > 0$ decreases $x(\delta(s))$, which can be possible by Lemma 2.5. The definition of $\epsilon_{a,b}$ indicates that splitting $\{sa, sb\}$ does not increase $\epsilon_{a',b'}$ for the other pairs $\{sa', sb'\}$ of edges. Hence the number of pairs to be split is at most $\binom{|V|-1}{2}$. From now on, we show that $\epsilon_{a,b}$ can be computed in strongly polynomial time. This means that the complete splitting at s can be executed in strongly polynomial time.

Lemma 2.6. *For an edge-weighted undirected graph (V, x) and $s, a, b \in V$, $\epsilon_{a,b}$ can be computed in polynomial time.*

Proof. In this proof, we show that

$$\epsilon_{a,b} = \min\{x(sa), x(sb)\} - \max_{u,v\in V-s}(\lambda(u, v; V, x) - \lambda(u, v; V, x''))/2 \tag{2.29}$$

holds, where x'' is the vector obtained by splitting $\{sa, sb\}$ by $\min\{x(sa), x(sb)\}$. This tells how to compute $\epsilon_{a,b}$ in strongly polynomial time because $\lambda(u, v; V, x)$ and $\lambda(u, v; V, x'')$ can be computed by max-flow algorithms.

Suppose that $\min\{x(sa), x(sb)\} \le q_x(a, b)$ (i.e., $\epsilon_{a,b} = \min\{x(sa), x(sb)\}$). Then (2.29) is obvious since by the definition of $\epsilon_{a,b}$, $\lambda(u, v; V, x) = \lambda(u, v; V, x'')$ holds for every $u, v \in V - s$ in this case. So let us consider the other case, i.e., $\epsilon_{a,b} = q_x(a, b) < \min\{x(sa), x(sb)\}$. Let X^*, u^* and v^* be those that attain the minimum of (2.22).

First, let us show that $\lambda(u^*, v^*; V, x) > \lambda(u^*, v^*; V, x'') = x''(\delta(X^*))$. If $\lambda(u^*, v^*; V, x) = \lambda(u^*, v^*; V, x'')$, then

$$x(\delta(X^*)) - 2\min\{x(as), x(bs)\} = x''(\delta(X^*)) \ge \lambda(u^*, v^*; V, x'') = \lambda(u^*, v^*; V, x)$$

holds. This implies that $q_x(a, b) = (x(\delta(X^*)) - \lambda(u^*, v^*; V, x))/2 \ge \min\{x(sa), x(sb)\}$, a contradiction. Hence $\lambda(u^*, v^*; V, x) > \lambda(u^*, v^*; V, x'')$ holds. On the other hand, $\lambda(u^*, v^*; V, x'') \le x''(\delta(X^*))$ holds since $u^* \in X^* \subseteq V - v^*$. Now suppose that $\lambda(u^*, v^*; V, x'') < x''(\delta(X^*))$. Then since $\lambda(u^*, v^*; V, x) > \lambda(u^*, v^*; V, x'')$, there exists another X' such that $x''(\delta(X')) = \lambda(u^*, v^*, V, x'')$, $a, b \in X' \subset V - s$, and X' contains exactly one of u^* and v^*. Let $(u', v') = (v^*, u^*)$ if $v^* \in X'$ and $(u', v') = (u^*, v^*)$ otherwise. Then it holds $x(\delta(X')) - \lambda(u', v'; V, x) = 0 < x(\delta(X^*)) - \lambda(u^*, v^*; V, x)$, which contradicts the choice of u^*, v^* and X^*. Hence we can see that $\lambda(u^*, v^*; V, x'') = x''(\delta(X^*))$ also holds.

Next, let us see that u^* and v^* attain the maximum in (2.29). For this, let y and z be two vertices in $V - s$ such that $\{y, z\} \ne \{u^*, v^*\}$. If $\lambda(y, z; V, x) - \lambda(y, z; V, x'') = 0$, then $\lambda(y, z; V, x) - \lambda(y, z; V, x'') \le \lambda(u^*, v^*; V, x) - \lambda(u^*, v^*; V, x'')$ is obvious because the right hand side is larger than 0. Hence suppose that $\lambda(y, z; V, x) - \lambda(y, z; V, x'') > 0$. In this case, there exists $X' \subset V - s$ such that $x''(\delta(X)) = \lambda(y, z; V, x'')$, $a, b \in X'$, and X' contains exactly one of y and z. From the choice of X^*, u^* and v^*, it follows that

$$x(\delta(X')) - \lambda(y, z; V, x) \ge x(\delta(X^*)) - \lambda(u^*, v^*; V, x). \tag{2.30}$$

By the definition of X', it holds that $x(\delta(X')) - 2\min\{x(sa), x(sb)\} = x''(\delta(X')) \ge \lambda(y, z; V, x'')$. We have already seen that $x(\delta(X^*)) - 2\min\{x(sa), x(sb)\} = x''(\delta(X^*)) = \lambda(u^*, v^*; V, x'')$. Combining these and (2.30) indicates that

$$\lambda(u^*, v^*; V, x) - \lambda(u^*, v^*; V, x'') \le \lambda(y, z; V, x) - \lambda(y, z; V, x''),$$

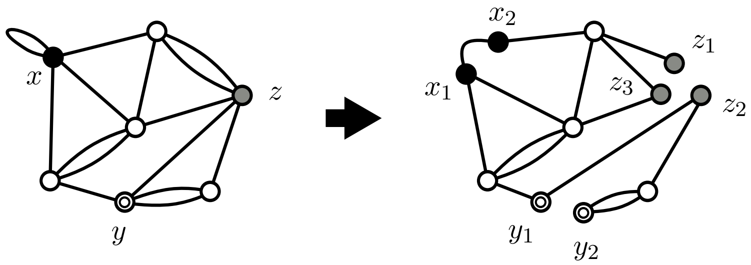


Figure 2.4: A detachment with $V_x = \{x_1, x_2\}$, $V_y = \{y_1, y_2\}$ and $V_z = \{z_1, z_2, z_3\}$ of an undirected graph

as required

From the above facts, we then have

$$\begin{aligned}
\epsilon_{a,b} = q_x(a,b) = \frac{x(\delta(X^*)) - \lambda(u^*, v^*; V, x)}{2} \\
= \frac{2\min\{x(sa), x(sb)\} + x''(\delta(X^*)) - \lambda(u^*, v^*; V, x)}{2} \\
= \min\{x(sa), x(sb)\} - \frac{\lambda(u^*, v^*; V, x) - \lambda(u^*, v^*; V, x'')}{2} \\
= \min\{x(sa), x(sb)\} - \max_{u,v \in V-s} \frac{\lambda(u, v; V, x) - \lambda(u, v; V, x'')}{2},
\end{aligned}$$

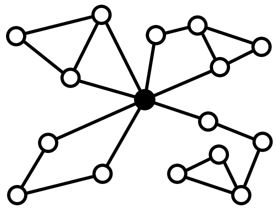
as required. $\square$

This lemma tells the strongly polynomiality of computing splittable pair in undirected multigraphs.

2.5.3 Detachment

For an undirected graph G, a *degree specification* $g = (\mathcal{V}, \rho)$ consists of a family $\mathcal{V} = \{V_v \mid v \in V\}$ of disjoint new vertex sets each of which corresponds to a vertex $v \in V$ and a function $\rho : V^* = \cup_{v \in V} V_v \to \mathbb{Z}_+$ such that $\sum_{x \in V_v} \rho(x) = \deg(v; G)$ for each $v \in V$. A *g-detachment* G^* of G is a graph obtained from G by replacing each $v \in V$ with vertices in V_v changing end vertices of each edge $uv \in E$ from u to some $x \in V_u$ (resp., from v to some $y \in V_v$) so that $\deg(z; G^*) = \rho(z)$ holds for each $z \in V^*$. This is a reverse operation of contraction since G is regained from G^* by contracting each V_v into a single vertex v. If $|V_v| = 1$ for all $v \in V - s$, then g may be especially denoted by $g(s) = \{V_s, \rho\}$ in order to emphasize the fact that only $s \in V$ is split into several vertices. Such $g(s)$-detachments may be especially referred to as *single detachments* in contrast to the general class of detachments, which is called *global detachments*. Figure 2.4 shows an example of detachments of an undirected graph, where vertices x, y and z are replaced by $V_x = \{x_1, x_2\}$, $V_y = \{y_1, y_2\}$ and $V_z = \{z_1, z_2, z_3\}$, respectively.

We note that the detachment generalizes splitting in the following sense. Let G^* be a $g(s)$-detachment of an undirected graph $G = (V, E)$ with $V_s = \{s, s'\}$, $\rho(s) = \deg(s; G) - 2$ and $\rho(s') = 2$ constructed by changing the end vertices of edges e and f from s to s'. Then the edge-connectivity $\lambda(u, v; G^*)$ between two vertices $u, v \in V$ is equal to $\lambda(u, v; G^{e,f})$. Removing s' by splitting $\{e, f\}$ transforms G^* into $G^{e,f}$. Figure 2.5 shows a detachment equivalent to $G^{e,f}$ in Figure 2.3.



G^*

Figure 2.5: A detachment G^* corresponding to $G^{e,f}$ in Figure 2.3

Historically detachments are introduced by C. St. J. A. Nash-Williams [62]. He showed a necessary and sufficient condition for the existence of k-edge-connected g-detachments. This result can be regarded as a generalization of the famous Euler's theorem, which shows the existence of Euler tours in Eulerian graphs; Euler's theorem tells the existence of 2-edge-connected g-detachments for Eulerian graphs, where $\rho(x) = 2$ for all $x \in V^*$. B. Fleiner [19] showed a necessary and sufficient condition for the existence of a $g(s)$-detachment that is k-edge-connected in $V - s$. His result was generalized by T. Jordán and Z. Szigeti [47] for the existence of $g(s)$-detachments that are r-edge-connected in $V - s$, which is formally stated as follows.

Theorem 2.15 ([47]). *Let $G = (V, E)$ be an undirected graph, $s \in V$ be a specified vertex to which no cut-edges are incident, and $g(s)$ be a degree specification consisting of V_s and $\rho : V_s \to \mathbb{Z}_+$. There exists a $g(s)$-detachment $G^* = (V^*, E^*)$ of G which is r-edge-connected in $V^* - V_s$ if and only if G is r-edge-connected in V–s and $\lambda(u, v; G - s) \geq r(u, v) - \sum_{x \in V_s} \lfloor \rho(x)/2 \rfloor$ holds for every pair $u, v \in V - s$.* □

Recently H. Nagamochi [61] considered the existence of loopless connected g-detachments and applied it to a graph inference problem.

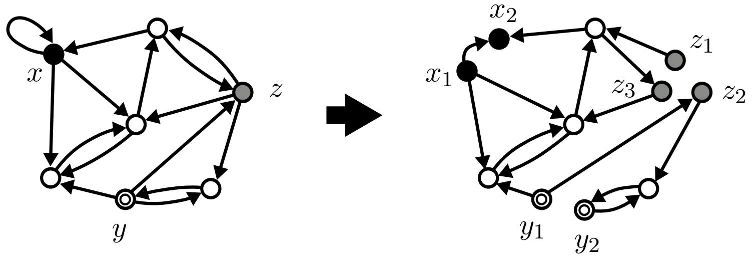


Figure 2.6: A detachment with $V_x = \{x_1, x_2\}$, $V_y = \{y_1, y_2\}$ and $V_z = \{z_1, z_2, z_3\}$ of a digraph

On the other hand, a degree specification $g = (\mathcal{V}, \rho^+, \rho^-)$ for a digraph D consists of $\mathcal{V} = \{V_v \mid v \in V\}$ and $\rho^+, \rho^- : V^* = \cup_{v \in V} V_v \to \mathbb{Z}_+$ such that $\sum_{x \in V_v} \rho^+(x) = \deg^+(v; D)$ and $\sum_{x \in V_v} \rho^-(x) = \deg^-(v; D)$. A g-detachment D^* of D is a digraph obtained from D by replacing each $v \in V$ with vertices in V_v changing end vertices of each arc $uv \in A$

from u to some $x \in V_u$ (resp., from v to some $y \in V_v$) so that $\deg^+(z; G^*) = \rho^+(z)$ and $\deg^-(z; G^*) = \rho^-(z)$ hold for each $z \in V^*$. Figure 2.6 shows an example of detachments of a digraph, where vertices x, y and z are replaced by $V_x = \{x_1, x_2\}$, $V_y = \{y_1, y_2\}$ and $V_z = \{z_1, z_2, z_3\}$, respectively. Analogously to undirected graphs, we represent g by $g(s)$ if $|V_v| = 1$ for $v \in V - s$.

As a result on detachments of digraphs, there is a counterpart of the Nash-Williams' result [62] due to A. R. Berg et al. [5].

Chapter 3

Network Design with Edge Dominating Constraints

In this chapter, we introduce some variants of the edge dominating set problem, which is one of the fundamental covering problems. Moreover, we propose approximation algorithms for these problems. The analysis of our algorithms is based on the relationship between polyhedra related to the problems.

3.1 Introduction

We consider an undirected graph $G = (V, E)$. An edge $e = uv \in E$ *dominates* edges incident to vertices u and v. An *edge dominating set* is defined as a set $F \subseteq E$ of edges such that each edge in E is dominated by at least one edge in F. Figure 3.1 illustrates an example of the edge dominating sets, which is displayed by dotted lines. The *edge dominating set problem* is formulated as follows.

> **Edge dominating set problem**
> Given a graph $G = (V, E)$ and an edge cost $w : E \to \mathbb{Q}_+$, find a minimum cost edge dominating set of G.

The edge dominating set problem is one of the fundamental covering problems and has some useful applications [3, 79]. It is known that the cardinality case of the edge dominating set problem is NP-hard even for some restricted classes of graphs such as planar or bipartite graphs of maximum degree 3 [79]. For the cardinality case, an arbitrary algorithm that outputs a maximal matching is a 2-approximation algorithm since there is a minimum edge dominating set which is also a maximal matching and any two maximal matchings $M_1, M_2 \subseteq E$ satisfy $|M_2|/2 \leq |M_1| \leq 2|M_2|$ [8, 39].

In general, the edge dominating set problem is shown to be approximable within factor of $2r$ if there is an r-approximation algorithm for the minimum cost vertex cover problem [8], where currently $r \leq 2$ is known. Furthermore, R. Carr et al. [8] presented a 2.1-approximation algorithm to this problem. His algorithm first constructs an instance of the minimum cost edge cover problem (introduced in Section 2.2) from the original instance and then finds

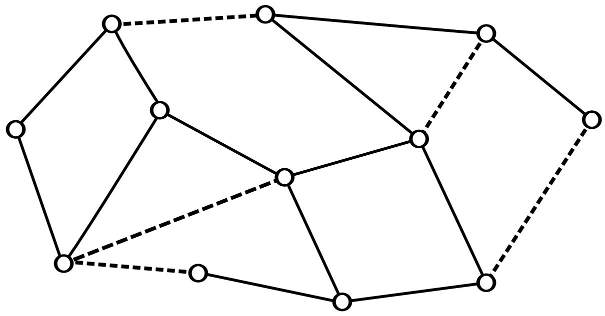


Figure 3.1: An example of the edge dominating sets (represented by dotted lines)

an optimal edge cover in the resulting instance. A key property for this method is that an edge cover in the resulting instance is also an edge dominating set for the original instance and that its cost is at most 2.1 times of the minimum cost of edge dominating sets in the original instance. The property is proved based on a relationship between the fractional edge dominating set polyhedron and the edge cover polyhedron $\mathrm{EC}(G, 1, +\infty)$ introduced in Section 2.2, where the fractional edge dominating set polyhedron contains all incidence vectors of edge dominating sets, but may not be the convex hull of these vectors. Afterward by using a refined edge dominating set polyhedron, T. Fujito and H. Nagamochi [26] gave a 2-approximation algorithm to the edge dominating set problem. Moreover, J. Könemann et al. proposed 3-approximation algorithms for two related problems; they ask to find a minimum cost edge dominating set which forms a tree/tours [50]. Note that, in the above algorithms, a linear programming relaxation of the integer programming formulation is used, but an output solution is not constructed directly from solutions of the linear programming or its dual problem. So it is an important issue to investigate whether such technique for designing approximation algorithms based on polyhedral structures can be extended to other problems.

There are also several results on the approximation hardness of the edge dominating set problem. It is known that the cardinality case of the edge dominating set is inapproximable within $\frac{7}{6} - \delta$ with an arbitrary $\delta > 0$ [12]. In general, there exists a reduction from approximation algorithms for the vertex cover problem to those for the edge dominating set problem while preserving their approximation factors [8]. The vertex cover problem is a well-studied covering problem and is proven to be inapproximable within any constant factor smaller than $10\sqrt{5} - 21 \approx 1.36067$ [14]. Therefore, the same result of the edge dominating set problem follows. In addition, it is widely believed that there is no approximation algorithm for the vertex cover problem whose constant approximation factor is less than 2, which indicates the difficulty to improve the 2-approximation algorithm for the edge dominating set problem.

On the other hand, a packing version of the edge dominating set problem is called *induced matching problem* (or *strong matching problem*). For an undirected graph $G = (V, E)$, a set $F \subseteq E$ of edges is called an *induced matching* (or *strong matching*) if the distance between any two edges in F is at least two, where the *distance* between two edges is defined as the shortest length of paths joining their end vertices. The induced matching problem is defined as follows.

> **Induced matching problem**
> Given an undirected graph $G = (V, E)$ and an edge cost $w : E \to \mathbb{Q}_+$, find an induced matching of the maximum cost.

This problem has some theoretical and practical applications such as irredundancy number [33], secure communication channels [34] and strong edge coloring [69]. By this reason, the induced matching problem has been studied well so far. L. J. Stockmeyer and V. V. Vazirani [74] introduced the notion of the induced matching and noted that it represents a kind of "risk-free marriage". They also introduced a *k-separated matching*, which is defined as a set $F \subseteq E$ of edges such that the distance between two edges in F is at least k (See also Figure 3.2). Obviously 1-separated matchings are ordinary matchings and 2-separated matchings are induced matchings. Then they showed that, for any k, the problem of finding a k-separated matching of the maximum cardinality is NP-hard in bipartite graphs of maximum degree 4, implying the NP-hardness of the maximum induced matching problem in the same class of graphs. Besides this, the NP-hardness of the maximum induced matching problem is proven for some other classes of graphs [7, 80, 49, 55] while some polynomially solvable classes are also discovered [7]. With regards to the approximation algorithms, APX-hardness is shown for r-regular graphs [80, 16]. In [80], M. Zito gave an approximation algorithm for those graphs, whose approximation factor is $r - (r-1)/(2r-1)$ while W. Duckworth et al. proposed an algorithm whose asymptotic approximation factor is $r - 1$ in [16]. For general graphs, the problem cannot be approximated within a constant factor unless P=NP [80].

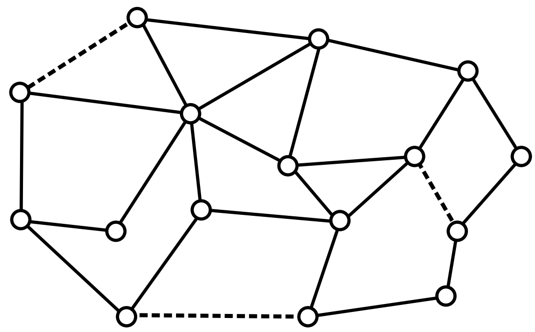


Figure 3.2: An example of 2-separated matchings, which is displayed by dotted lines

In this chapter, we introduce several natural extensions of the edge dominating set, the induced matching, and the edge cover, which will be expected to provide more flexible modelings in practice. We present approximation algorithms for those problems by investigating polyhedral structures of the convex hulls of their feasible solutions.

3.2 Capacitated b-edge dominating set problem

In this section, we generalize the edge dominating set problem by introducing capacities and demands on the edges. For a demand $b : E \to \mathbb{Z}_+$ of dominating edges and a capacity $c : E \to \mathbb{Z}_+$ of multiplicity, a *capacitated b-edge dominating set* ((b, c)-EDS) F is a set of edges such that each $e \in E$ is dominated by at least $b(e)$ edges in F, where F is allowed to contain at most $c(e)$ multiple copies of edge e. The *capacitated b-edge dominating set problem* is defined as follows.

Capacited b-edge dominating set ((b,c)-EDS) problem
Given a graph $G=(V,E)$, a demand $b: E \to \mathbb{Z}_+$ of dominating edges, a capacity $c: E \to \mathbb{Z}_+$ of multiplicity, and an edge cost $w: E \to \mathbb{Q}_+$, find a minimum cost (b,c)-EDS.

If $c=+\infty$, then (b,c)-EDS problem is especially called a *b-edge dominating set (b-EDS) problem.*

For an instance $(G=(V,E),b,c,w)$, an integer programming of the (b,c)-EDS problem is given as

$$\begin{array}{lll} \text{minimize} & w^T x & \\ \text{subject to} & x(e) \le c(e) & \text{for each } e \in E, \\ & x(\delta(e)) \ge b(e) & \text{for each } e \in E, \\ & x \in \mathbb{Z}_+^E. & \end{array} \tag{3.1}$$

Let $\mathrm{EDS}(G,b,c)$ denote the feasible region of the linear programming obtained by relaxing the integrality constraints in problem (3.1). That is to say, $\mathrm{EDS}(G,b,c)$ is the set of vectors $x \in \mathbb{R}_+^E$ such that

$$0 \le x(e) \le c(e) \qquad \text{for each } e \in E, \tag{3.2a}$$

$$x(\delta(e)) \ge b(e) \qquad \text{for each } e \in E. \tag{3.2b}$$

The minimum cost of vectors in $\mathrm{EDS}(G,b,c)$ is a lower bound on the minimum cost of a given instance (G,b,c,w) for the (b,c)-EDS problem.

Now we present an approximation algorithm for the (b,c)-EDS problem. Given an instance (G,b,c,w) of the (b,c)-EDS problem, the algorithm first constructs an instance of the (a,c)-edge cover problem (see Section 2.2.3) and then computes an optimal solution for it as an approximate solution to the input instance. The algorithm needs a parameter $f>0$. This parameter has no effect on the feasibility of solutions that the algorithm outputs. However, it must be set to an appropriate value for achieving a good approximation factor when $c(e)$ is finite for some $e \in E$ as described later. A formal description of our algorithm is the following.

Algorithm DOMINATE(f)

Input: An undirected graph $G=(V,E)$, a demand $b: E \to \mathbb{Z}_+$ of dominating edges, a capacity $c: E \to \mathbb{Z}_+$ of multiplicity, an edge cost $w: E \to \mathbb{Q}_+$, and a real $f>0$

Output: A (b,c)-EDS or "INFEASIBLE"

1: **if** $\mathrm{EDS}(G,b,c)=\emptyset$ **then**
2: output "INFEASIBLE" and halt
3: **end if;**
4: Compute $x^* \in \mathrm{EDS}(G,b,c)$ that minimizes $w^T x$;

$E' := \emptyset$;
for each $e \in E$ with $fx^*(e) > c(e)$ **do**
    $\bar{x}(e) := c(e)$; $E' := E' \cup \{e\}$;
    for each $e' \in \delta(e)$ **do**
        $b(e') := \max\{0, b(e') - c(e)\}$
    end for
end for;
for each edge $e = (u, v) \in E$ **do**
    $b'_{x^*}(u, e) := b(e)$;
    if $x^*(\delta(u) - E') \geq x^*(\delta(v) - E')$ **then**
        $b'_{x^*}(v, e) := 0$; $b'_{x^*}(u, e) := b(e)$;
    else
        $b'_{x^*}(u, e) := 0$; $b'_{x^*}(v, e) := b(e)$;
    end if
end for
for each vertex $v \in V$ **do**
    $a_{x^*}(v) := \max_{e \in \delta(v)} b'_{x^*}(v, e)$
end for;
Compute a minimum cost (a_{x^*}, c')-edge cover $\bar{x}_{E-E'}$ for $G' = (V, E - E')$, $c' = c_{E-E'}$ and $w' = w_{E-E'}$;
Output $\bar{x}$ as a (b, c)-EDS to (G, b, c, w).

If the input instance is infeasible, then there exists an edge $e \in E$ with $c(\delta(e)) < b(e)$. In this case, $\mathrm{EDS}(G, b, c) = \emptyset$ holds and DOMINATE(f) outputs "INFEASIBLE". In the following, we suppose that the given instance is feasible, and hence $\mathrm{EDS}(G, b, c) \neq \emptyset$.

We first show that $\bar{x}$ is a (b, c)-EDS. For an edge $e = (u, v) \in E$, let us suppose $x^*(\delta(u) - E') \geq x^*(\delta(v) - E')$. Then

$$\bar{x}(\delta(u) - E') \geq a_{x^*}(u) \geq b(e) - c(\delta(e) \cap E').$$

The above first inequality holds since $\bar{x}_{E-E'}$ is an (a_{x^*}, c')-edge cover, and the second one holds by the definition of a_{x^*}. Since $\bar{x}(\delta(e) \cap E') = c(\delta(e) \cap E')$, it holds

$$\bar{x}(\delta(e)) \geq \bar{x}(\delta(u) - E') + \bar{x}(\delta(e) \cap E') \geq b(e).$$

We can easily check that $0 \leq \bar{x}(e) \leq c(e)$ also holds. Hence, $\bar{x}$ is a (b, c)-EDS and algorithm DOMINATE(f) outputs a feasible solution.

We now analyze the approximation factor of algorithm DOMINATE(f) by establishing a relationship between $\mathrm{EDS}(G, b, c)$ and $\mathrm{EC}(G, a_{x^*}, c')$. In the following discussion, we suppose that $b(e) \geq 1$ for at least one edge $e \in E$, since if $b(e) = 0$ for all edges $e \in E$, DOMINATE(f) apparently outputs the optimal solution $\bar{x} = 0^E$. At first, we consider the b-EDS problem, i.e., $c = +\infty$. In this case, the parameter f makes no effect on the choice of E' in the algorithm and $E' = \emptyset$ always holds.

Lemma 3.1. *Let x be a vector in* $\mathrm{EDS}(G = (V, E), b, +\infty)$*, and $a_x : V \to \mathbb{Z}_+$ be the demand constructed from x in algorithm DOMINATE(f). Then vector $2x \in \mathbb{R}_+^E$ satisfies conditions* (2.5a) *and* (2.5b) *for* $\mathrm{EC}(G, a_x, +\infty)$.

Proof. Let $x \in \mathrm{EDS}(G, b, +\infty)$. Then vector $2x$ satisfies condition (2.5a) for $\mathrm{EC}(G, a_x, +\infty)$ because $x \in \mathbb{R}_+^E$ holds by (3.2a) for $\mathrm{EDS}(G, b, +\infty)$. We now show that $2x$ satisfies (2.5b), i.e., $2x(\delta(v)) \geq a_x(v)$ for all $v \in V$. Let v be a vertex in V. Then there is an edge $e = (u, v) \in E$ such that $a_x(v) = b'_x(v, e)$. If $b'_x(v, e) = 0$, then we have $2x(\delta(v)) \geq 0 = a_x(v)$ since $x \in \mathbb{R}_+^E$ holds. Therefore, let us assume $b'_x(v, e) > 0$. Then $b'_x(v, e) = b(e)$ and $x(\delta(v)) \geq x(\delta(u))$ hold. Now $x(\delta(e)) \geq b(e)$ holds by (3.2b) for $\mathrm{EDS}(G, b, +\infty)$, which implies $x(\delta(v)) + x(\delta(u)) = x(\delta(e)) + x(e) \geq b(e) + x(e)$ holds. Then we have

$$2x(\delta(v)) \geq x(\delta(u)) + x(\delta(v)) \geq b(e) + x(e) \geq b(e) = b'_x(v, e) = a_x(v).$$

Therefore, (2.5b) also holds for $2x$. □

Lemma 3.2. *For a simple undirected graph $G = (V, E)$ and a demand $a : V \to \mathbb{Z}_+$, let $\beta = \min\{a(v) \mid v \in V, a(v) \neq 0\}$. Then, for any vector $x' \in \mathbb{R}_+^E$ satisfying conditions* (2.5a) *and* (2.5b) *for* $\mathrm{EC}(G, a, +\infty)$*, the vector*

$$y = \left(1 + \frac{1}{2\lfloor 3\beta/2 \rfloor + 1}\right) x' \in \mathbb{R}_+^E$$

satisfies condition (2.5c') *for* $\mathrm{EC}(G, a, +\infty)$.

Proof. Let U be a subset of V such that $a(U)$ is odd. It suffices to show that (2.5c') holds for $x = y$ and U. If U contains a vertex v such that $a(v) = 0$, then (2.5c') follows inductively from $y(E[U']) + y(\delta(U')) \geq \lceil a(U')/2 \rceil$ for $U' = U - \{v\}$, since $y(E[U]) + y(\delta(U)) \geq y(E[U']) + y(\delta(U'))$ and $a(U) = a(U')$. Hence we assume without loss of generality that $a(v) \geq \beta$ for all $v \in U$. Moreover, if $|U| = 1$, then (2.5c') is implied by (2.5b) since for $U = \{v\}$, $y(E[U]) + y(\delta(U)) = y(\delta(v)) \geq x'(\delta(v)) \geq a(v) \geq \lceil a(v)/2 \rceil$. We now consider the case of $|U| = 2$. Let $U = \{v_1, v_2\}$. Since $a(U) = a(v_1) + a(v_2)$ is odd, $a(v_1) \neq a(v_2)$ holds, where we assume without loss of generality $a(v_1) > a(v_2)$. Then

$$\left\lceil \frac{a(U)}{2} \right\rceil = \left\lceil \frac{a(v_1) + a(v_2)}{2} \right\rceil \leq a(v_1).$$

We have

$$x'(E[U]) + x'(\delta(U)) \geq x'(\delta(v_1))$$

because $E[U] \cup \delta(U) \supseteq \delta(v_1)$. Since x' satisfies $x'(\delta(v_1)) \geq a(v_1)$ by (2.5b), we have

$$\begin{aligned} y(E[U]) + y(\delta(U)) &\geq x'(E[U]) + x'(\delta(U)) \\ &\geq x'(\delta(v_1)) \geq a(v_1) \geq \left\lceil \frac{a(v_1) + a(v_2)}{2} \right\rceil = \left\lceil \frac{a(U)}{2} \right\rceil. \end{aligned}$$

In what follows, we assume that $|U| \geq 3$ and $a(v) \geq \beta$ for all $v \in U$. Since $x'(\delta(v)) \geq a(v)$ holds for all $v \in U$ by (2.5b) for $\mathrm{EC}(G, a, +\infty)$, we have

$$2x'(E[U]) + x'(\delta(U)) = \sum_{v \in U} x'(\delta(v)) \geq a(U).$$

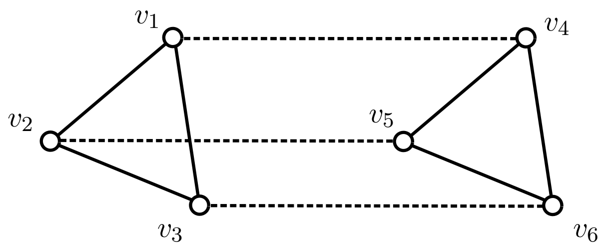


Figure 3.3: A tight example for the analysis for the performance of DOMINATE(f)

Therefore

$$x'(E[U]) + x'(\delta(U)) \geq \frac{a(U) + x'(\delta(U))}{2} \geq \frac{a(U)}{2}.$$

To show (2.5c'), we only have to prove that

$$\frac{\lceil a(U)/2 \rceil}{a(U)/2} = 1 + \frac{1}{a(U)} \leq 1 + \frac{1}{2\lfloor 3\beta/2 \rfloor + 1},$$

or equivalently

$$a(U) \geq 2\lfloor 3\beta/2 \rfloor + 1. \tag{3.3}$$

From the assumption, $a(U) \geq 3\beta$ holds. Moreover, since $a(U)$ is odd, $a(U) \geq 3\beta + 1$ if 3β is even. This implies (3.3). □

Theorem 3.1. *Let $\beta = \min\{b(e) \mid e \in E, b(e) \neq 0\}$. Algorithm DOMINATE($f$) delivers an approximate solution of a cost within a factor of*

$$\rho = 2\left(1 + \frac{1}{2\lfloor 3\beta/2 \rfloor + 1}\right)\left(\leq \frac{8}{3}\right)$$

to the b-EDS problem.

Proof. Let $\bar{x} \in \mathbb{Z}_+^E$ be a vector obtained by algorithm DOMINATE(f). We have already observed that $\bar{x}$ is a $(b, +\infty)$-EDS to instance $(G, b, +\infty, w)$. We show that $\bar{x}$ is a ρ-approximate solution. We denote by OPT the minimum cost of a $(b, +\infty)$-EDS for $(G, b, +\infty, w)$. Let $x^* \in \mathbb{R}_+^E$ be the vector minimizing $w^T x^*$ over EDS$(G, b, +\infty)$, which is computed in DOMINATE(f). Since EDS$(G, b, +\infty)$ contains a minimum cost $(b, +\infty)$-EDS, it holds $w^T x^* \leq$ OPT. By Lemma 3.1, vector $2x^*$ satisfies conditions (2.5a) and (2.5b) for EC$(G, a_{x^*}, +\infty)$. Since $b(e) \geq \beta$ for all $e \in E$ such that $b(e) \neq 0$, we see that $a_{x^*}(v) \geq \beta$ or $a_{x^*}(v) = 0$ holds for each $v \in V$. Therefore, from Lemma 3.2, we have $\rho x \in$ EC$(G, a_{x^*}, +\infty)$. Since algorithm DOMINATE(f) outputs a solution $\bar{x}$ of minimum cost over all vectors in EC$(G, a_{x^*}, +\infty)$, we have $w^T \bar{x} \leq \rho w^T x^*$, from which $w^T \bar{x} \leq \rho$OPT follows, as required. □

Figure 3.3 shows an instance $(G, b, +\infty, w)$ that indicates that the analysis of Theorem 3.1 is tight in the case of $\beta = 1$. The instance consists of an undirected graph $G = (V = \{v_1, \ldots, v_6\}, E = E_1 \cup E_2\})$, a demand b and an edge cost w such that

$$b(e) = \begin{cases} 0 & \text{if } e \in E_1, \\ 1 & \text{if } e \in E_2, \end{cases}$$

$$w(e) = \begin{cases} 1 & \text{if } e \in E_1, \\ +\infty & \text{if } e \in E_2, \end{cases}$$

where in Figure 3.3, edges in $E_1 = \{v_1v_2, v_2v_3, v_1v_3\}$ and $E_2 = \{v_4v_5, v_5v_6, v_4v_6\}$ are represented by solid lines and by dotted lines, respectively. For this instance, DOMINATE(f) computes x^* such that

$$x^*(e) = \begin{cases} \frac{1}{4} & \text{if } e \in E_1, \\ 0 & \text{if } e \in E_2, \end{cases}$$

for which $a_{x^*}(v_1) = a_{x^*}(v_2) = a_{x^*}(v_3) = 1$ holds. Then we need to multiply x^* by 8/3 so that $y = (8/3)x^*$ satisfies (2.5c') for $U = \{v_1, v_2, v_3\}$. Therefore this instance $(G, b, +\infty, w)$ achieves $\rho = 8/3$, implying that our analysis $w^T x^*$ as a lower bound on the optimal value is tight. Note, however, that the algorithm DOMINATE(f) outputs an optimal solution $\bar{x}$ to this instance. To show the optimality in this case, we need to discover a stronger lower bound.

In addition, algorithm DOMINATE(f) achieves a better approximation factor in some special cases. We introduce some results.

Theorem 3.2. *For a demand $b : E \to \mathbb{Z}_+$ such that $\beta = \min_{e \in E} b(e) \geq 1$, algorithm DOMINATE($f$) delivers an approximate solution of a cost within a factor of*

$$\rho = 2\left(1 + \frac{1}{4\beta + 1}\right) \left(\leq \frac{12}{5}\right)$$

to the b-EDS problem.

Proof. Let $x \in \mathrm{EDS}(G, b, +\infty)$ and U be a subset of V such that $|U| \geq 3$ and $a_x(U) < 4\beta + 1$, where $a_x : E \to \mathbb{Z}_+$ is the demand constructed from x in DOMINATE(f). Below we show that the vector $y = 2x$ satisfies (2.5c') for $\mathrm{EC}(G, a_x, +\infty)$ and U. From this fact, we can assume without loss of generality that $b(U) \geq 4\beta + 1$. Combined with Lemma 3.1 and the discussion in the proof of Lemma 3.2, this proves the theorem.

Let $e \in E[U]$. Then it holds $x(E[U]) + x(\delta(U)) \geq x(E[U]) \geq x(\delta(e)) \geq b(e) \geq \beta$. Therefore, $y(E[U]) + y(\delta(U)) \geq 2\beta$. On the other hand, we have $\lceil a_x(U)/2 \rceil \leq 2\beta$ from the assumption. Combining these inequalities leads to $y(E[U]) + y(\delta(U)) \geq \lceil a_x(U)/2 \rceil$, as required. □

Theorem 3.3. *Algorithm DOMINATE(f) is a 2-approximation algorithm for the b-EDS problem in bipartite graphs.*

Proof. For bipartite graphs, the edge cover polytopes are determined by only inequalities (3.2a) and (3.2b) [71]. Hence the theorem follows from Lemma 3.1. □

When b takes the same value for all edges, a better guarantee can be derived as follows.

Lemma 3.3. *Let $x \in \mathbb{R}_+^E$ be a vector in $\mathrm{EDS}(G, b, +\infty)$. If $b(e) = \beta \geq 1$ for all $e \in E$, then ρx belongs to $\mathrm{EC}(G, a_x, +\infty)$, where $\rho = 2.1$ for $\beta = 1$ and $\rho = 2$ for $\beta \geq 2$.*

Proof. Lemma 3.1 shows that $2x$ satisfies (2.5a) and (2.5b) for $\mathrm{EC}(G, a_x, +\infty)$. Therefore, it suffices to prove that ρx satisfies (2.5c') for $\mathrm{EC}(G, a_x, +\infty)$. Let U be a subset of V such that $a_x(U)$ is odd. As in the proof of Lemma 3.2, we can assume that $|U| \geq 3$ and $a_x(v) \geq \beta$ holds for all $v \in U$.

Let $x' = 2x$. From the inequalities (2.5b) for $\mathrm{EC}(G, a_x, +\infty)$ and (3.2b) for $\mathrm{EDS}(G, b, +\infty)$ we get that

$$x'(\delta(u)) + x'(\delta(v)) \geq \begin{cases} 2b(e) + x'(e) & e = (u, v) \in E, \\ a_x(u) + a_x(v) & \text{otherwise.} \end{cases}$$

By summing up the above inequalities over all pairs of distinct u and v in $U \times U$, we get

$$\begin{aligned} (|U| - 1) \sum_{u \in U} x'(\delta(u)) &\geq 2b(E[U]) + x'(E[U]) + \sum_{\substack{u,v \in U \\ (u,v) \notin E}} (a_x(u) + a_x(v)), \\ &= 2b(E[U]) + x'(E[U]) + (|U| - 1)a_x(U) - \sum_{\substack{u,v \in U \\ (u,v) \in E}} (a_x(u) + a_x(v)). \end{aligned}$$

Now, $b(e) = \beta$ for all $e \in E$. Hence $a_x(v) \leq \beta$ for each $v \in V$. This leads to $2b(e) \geq a_x(u) + a_x(v)$ for each $e = (u, v) \in E$, which implies

$$2b(E[U]) - \sum_{\substack{u,v \in U \\ (u,v) \in E}} (a_x(u) + a_x(v)) \geq 0.$$

Therefore, we have

$$(|U| - 1) \sum_{u \in U} x'(\delta(u)) \geq x'(E[U]) + (|U| - 1)a_x(U).$$

Recall that $|U| \geq 3$ is assumed. Note that $\sum_{u \in U} x'(\delta(u)) = x'(\delta(U)) + 2x'(E[U])$. Hence

$$x'(E[U]) + x'(\delta(U)) \geq \frac{(|U| - 2)x'(\delta(U)) + (|U| - 1)a_x(U)}{2|U| - 3} \geq \frac{(|U| - 1)a_x(U)}{2|U| - 3}.$$

Therefore, we have

$$\begin{aligned} \frac{\lceil a_x(U)/2 \rceil}{x'(E[U]) + x'(\delta(U))} &= \frac{(a_x(U) + 1)/2}{(|U| - 1)a_x(U)/(2|U| - 3)} \\ &= \left(1 + \frac{1}{a_x(U)}\right) \cdot \frac{2|U| - 3}{2|U| - 2}. \end{aligned} \tag{3.4}$$

We analyze the maximum value of the right hand size of (3.4). Since we consider the case where $a_x(v) \geq \beta$ holds for all $v \in U$, it holds that $a_x(U) \geq \beta|U|$. Therefore we have

$$\left(1 + \frac{1}{a_x(U)}\right) \cdot \frac{2|U| - 3}{2|U| - 2} \leq \left(1 + \frac{1}{\beta|U|}\right) \cdot \frac{2|U| - 3}{2|U| - 2}. \tag{3.5}$$

For $\beta = 1$, the right hand side of (3.5) takes the maximum value $21/20$ when $|U| = 5$. On the other hand, if $\beta \geq 2$, then the right hand side of (3.4) is at most 1. Therefore, ρx satisfies (2.5c') for $\mathrm{EC}(G, a_x)$. $\square$

Lemma 3.3 directly implies the following theorem.

Theorem 3.4. *Suppose that* $b(e) = \beta$ *for all* $e \in E$. *Then algorithm DOMINATE*(f) *delivers an approximate solution of a cost within a factor of* 2.1 *if* $\beta = 1$ *or a factor of* 2 *if* $\beta \geq 2$ *to the b-EDS problem.* □

We now analyze the approximation factor of DOMINATE(f) for the general (b, c)-EDS problem, i.e., when c takes finite values for some edges. In this case, we need to set f to an appropriate value. Let $\beta = \min\{a_x(U) - c(F) \mid U \subseteq V, F \subseteq \delta(U) - E', a_x(U) - c(F)$ is odd and $\geq 3\}$ and $\rho = 2(1 + 1/\beta)$ be the factor. If $f \geq \rho$, we can prove that $\rho x_{E-E'} \in \mathrm{EC}(G' = (V, E - E'), a_x, c)$, where $x \in \mathrm{EDS}(G, b, c)$ (the proof is similar with that of Lemmas 3.1 and 3.2). Then, algorithm DOMINATE(f) achieves the approximation factor of f because of the following reasons. The cost of output edges in E' is bounded as

$$w_{E'}^T \bar{x}_{E'} \leq w_{E'}^T c_{E'} < f w_{E'}^T x_{E'}^*.$$

With regard to edges in $E - E'$, it holds that

$$w_{E-E'}^T \bar{x}_{E-E'} \leq \rho w_{E-E'}^T x_{E-E'}^*$$

from the above-mentioned relation. Hence, it holds

$$w^T \bar{x} = w_{E'}^T \bar{x}_{E'} + w_{E-E'}^T \bar{x}_{E-E'} < f w^T x^* \leq f\mathrm{OPT},$$

where OPT denotes the cost of the optimal solution. Notice that ρ depends on f because f decides which edges are added to E'. As we make f smaller while keeping $f \geq \rho$, we can obtain a better approximation factor. Especially, DOMINATE(8/3) is a 8/3-approximation algorithm.

Theorem 3.5. *DOMINATE*(8/3) *is an* 8/3*-approximation algorithm for the* (b, c)*-EDS problem.* □

We also obtain the same results described in Theorems 3.3.

Theorem 3.6. *DOMINATE*(2) *is a* 2*-approximation algorithm for the* (b, c)*-EDS problem in bipartite graphs.* □

3.3 Capacitated induced matching problem

In Section 3.2, the edge dominating set problem was generalized into the (b, c)-EDS problem by introducing edge capacities. In this section, we generalize the induced matching problem similarly. Before this, let us see a result about the approximation hardness of k-separated matching problem, which is another generalization of the induced matching problem.

3.3.1 Approximation hardness of the k-separated matching problem

First, let us formulate the k-separated matching problem formally.

> **k-separated matching problem**
> Given an undirected graph $G = (V, E)$, an edge cost $w : E \to \mathbb{Q}_+$ and an integer $k \geq 1$, find a maximum cost k-separated matching.

For an undirected graph $G = (V, E)$, a set $U \subseteq V$ is called an *independent set* if no edge in E joins vertices in U.

> **Independent set problem**
> Given an undirected graph $G = (V, E)$ and an edge cost $w : V \to \mathbb{Q}_+$, find a maximum cost independent set.

The independent set problem is known to be NP-hard [28] and admits no approximation algorithm whose approximation factor is better than $|V|^{\frac{1}{2}-\varepsilon}$ for any $\varepsilon > 0$ if P $\neq$ NP [40]. Furthermore, it is not approximable within $|V|^{1-\varepsilon}$ for any $\varepsilon > 0$ if P $\neq$ NP and NP $\neq$ ZPP [40].

In the following, we give a polynomial time reduction from the independent set problem to the k-separated matching problem, which implies the constant factor approximation hardness of the k-separated matching problem. It has been already shown in [80] that the independent set problem can be reduced to the induced matching problem (i.e., $k = 2$). We extend this to the k-separated matching problem with arbitrary $k \geq 2$.

Now we describe the reduction. First, we consider the case in which k is even. Let $k' = k/2$. For each vertex $v \in V$, prepare a set $V_v = \{v_1, \ldots, v_{k'}\}$ of new vertices and $E_v = \{vv_1, v_1v_2, \ldots, v_{k'-1}v_{k'}\}$ of new edges. Define $G^* = (V^*, E^*)$ as a graph such that $V^* = V \cup_{v \in V} V_v$ and $E^* = E \cup_{v \in V} E_v$. Then we can assume without loss of generality that a k-separated matching $M \subseteq E^*$ of G^* consists of edges in $\{v_{k'-1}v_{k'} \mid v \in V\}$ since if M contains an edge in $E_v - \{v_{k'-1}v_{k'}\}$ or $uv \in E$, we can replace it by $v_{k'-1}v_{k'}$ while preserving the feasibility and the cardinality of M. For a k-separated matching M in G^*, we can construct a corresponding independent set $U = \{v \in V \mid v_{k'-1}v_{k'} \in M\}$ in G; Actually U is an independent since, if $u, v \in U$ is adjacent, the distance between $u_{k'-1}u_{k'}$ and $v_{k'-1}v_{k'}$ is $2(k'-1)+1 = k-1$, contradicting to the fact that M is a k-separated matching. Moreover $|M| = |U|$. We can also immediately see the opposite direction of the correspondence.

In the next, let us consider the case in which k is odd. In this case, we let $k' = \lfloor k/2 \rfloor$, and prepare V_v and E_v for each $v \in V$ analogously to the case of even k. In addition, we subdivide each edge $e \in E$ with a new vertex z_e (i.e., replace $e = uv$ by uz_e and z_ev). Let $G^* = (V^*, E^*)$ be a graph such that $V_E = \{z_e \mid e \in E\}$, $E' = \{uz_e, z_ev \mid uv \in E\}$, $V^* = V \cup_{v \in V} V_v \cup V_E$, and $E^* = E' \cup E_v$. Figure 3.4 describes an example with $k = 5$, where black circles (resp., gray circles) denote in $\cup_{v \in V} V_v$ (resp., in V_E) and gray lines represent edges in E_v. We can assume without loss of generality that a k-separated matching $M \subseteq E^*$ of G^* consists of edges in $\{v_{k'-1}v_{k'} \mid v \in V\}$ since if M contains an edge in $E_v - \{v_{k'-1}v_{k'}\}$ or vz_e, we can replace it by $v_{k'-1}v_{k'}$ analogously to the previous case. Hence, we can construct a corresponding independent set $U = \{v \in V \mid v_{k'-1}v_{k'} \in M\}$ in G again. Accordingly, the transformation gives a reduction from the independent set to the k-separated matching with arbitrary k preserving the approximation factor.

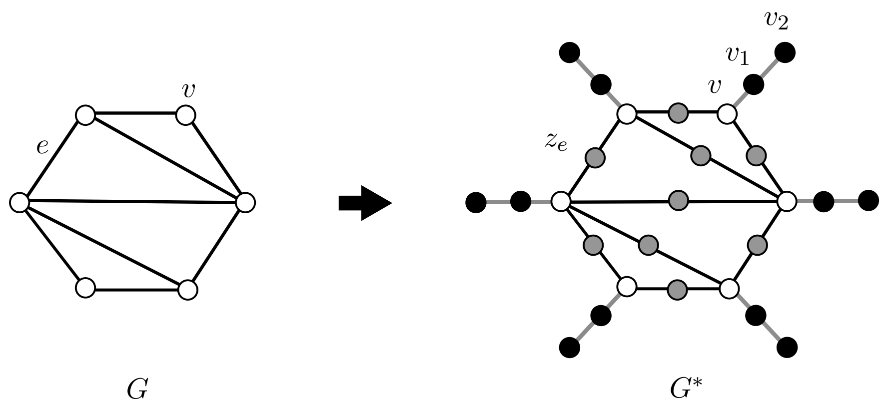


Figure 3.4: Reduction from the independent set problem to the 5-separated matching problem

Theorem 3.7. *If* $\mathrm{P} \neq \mathrm{NP}$*, the* k*-separated matching problem with some* $k \geq 2$ *is not approximable within* $|V|^{\frac{1}{2}-\varepsilon}$ *for any* $\varepsilon > 0$*. In addition, if* $\mathrm{NP} \neq \mathrm{ZPP}$*, then it is not approximable within* $|V|^{1-\varepsilon}$*.* □

3.3.2 Approximation algorithm for the capacitated induced matching problem

We now generalize the induced matching problem so that multiple edges between two vertices are allowed to be chosen, introducing two capacities $b, c : E \to \mathbb{Z}_+$ for a graph $G = (V, E)$. A multiset F of edges is called an (b, c)*-induced matching* if each edge $e \in E$ is dominated by at most $b(e)$ edges in F and the number of copies of $e \in E$ contained in F is at most $c(e)$.

(b, c)-induced matching problem
Given an undirected graph $G = (V, E)$, a capacity $b : E \to \mathbb{Z}_+$ of dominating edges, a capacity $c : E \to \mathbb{Z}_+$ of multiplicity, and an edge cost $w : E \to \mathbb{Q}_+$, find a maximum cost (b, c)-induced matching.

Since the (b, c)-induced matching problem contains the induced matching problem, the hardness of the induced matching problem described in Section 3.1 is carried over to this generalized problem, i.e., it cannot be approximated within a constant factor for general graphs. However, we show in the following that, if each capacity $b(e)$ is restricted to be larger than one, then it can be approximated within factor of $2/9$. This is an interesting fact because the k-separated matching problem, another generalization of the induced matching problem, is inapproximable within a constant factor for any k as already observed in Section 3.3.1.

Now we formulate the (b, c)-induced matching problem as the following integer programming.

$$\begin{array}{lll} \text{maximize} & w^T x & \\ \text{subject to} & x(\delta(e)) \leq b(e) & \text{for each } e \in E, \\ & x(e) \leq c(e) & \text{for each } e \in E, \\ & x \in \mathbb{Z}_+^E. & \end{array} \tag{3.6}$$

Let $\mathrm{IM}(G, b, c)$ be the set of vectors $x \in \mathbb{R}_+^E$ such that

$$0 \le x(e) \le c(e) \qquad \text{for each } e \in E, \tag{3.7a}$$

$$x(\delta(e)) \le b(e) \qquad \text{for each } e \in E. \tag{3.7b}$$

Observe that $\mathrm{IM}(G, b, c)$ represents the feasible region of the linear programming obtained from problem (3.6) by relaxing its integrality constraints. Although $\mathrm{IM}(G, b, c)$ contains all feasible solutions of (3.6), the set of optimal solutions over the region may include no integer solution for a given objective function.

To construct an approximate solution to a given instance (G, b, c) of the (b, c)-induced matching problem, we solve an instance (G, a, c) of the (a, c)-matching problem (see Section 2.2.4). The capacity vector a will be defined so that a (a, c)-matching is also a (b, c)-induced matching in G. The algorithm is described as follows.

Algorithm PACK

Input: An undirected graph $G = (V, E)$, capacity $b : E \to \mathbb{Z}_+$ of dominating edges, capacity $c : E \to \mathbb{Z}_+$ of multiplicity, and an edge cost $w : E \to \mathbb{Q}_+$

Output: A (b, c)-induced matching

for each $e = (u, v) \in E$ **do**
    $b'(e, u) := \lfloor b(e)/2 \rfloor$; $b'(e, v) := \lceil b(e)/2 \rceil$
end for;
for each $v \in V$ **do**
    $a(v) := \min_{e \in \delta(v)} b'(e, v)$
end for;
Output a maximum cost (a, c)-matching $\bar{x} \in \mathbb{Z}_+^E$ for G and w as a (b, c)-induced matching;

Integer vectors $x \in \mathbb{Z}^E$ satisfying (2.7a) and (2.7b) of $\mathrm{MA}(G, a, c)$ are (b, c)-induced matchings because $x(\delta(e)) \le x(\delta(u)) + x(\delta(v)) \le a(u) + a(v) \le b(e)$, In the following, we analyze the approximation factor of algorithm PACK. If $b(e) = 0$ for all $e \in E$, algorithm PACK obviously outputs the optimal solution. Hence in what follows, we suppose that at least one edge e satisfies $b(e) > 0$.

Lemma 3.4. *Let $x \in \mathrm{IM}(G, b, c)$ and $a : V \to \mathbb{Z}_+$ be a function obtained in algorithm PACK. Then vector*

$$x' = \frac{1}{2}\left(1 - \frac{1}{\beta_1}\right) x$$

satisfies conditions (2.7a) *and* (2.7b) *for* $\mathrm{MA}(G, a, c)$*, where* $\beta_1 = \min\{b(e) \mid e \in E,\ b(e) \text{ is odd}\}$ *if there exists an edge e with odd $b(e)$, and $\beta_1 = +\infty$ otherwise.*

Proof. Since $x \in \mathrm{IM}(G, b, c)$ satisfies $0 \le x(e) \le c(e)$ for each $e \in E$, it is immediate to see that x' satisfies (2.7a) for $\mathrm{MA}(G, a, c)$. Then, we show that $x'(\delta(v)) \le a(v)$ holds for each $v \in V$.

Let $v \in V$. There is an edge $e = (u, v) \in E$ such that $a(v) = b'(e, v)$. Note that $x(\delta(v)) \le x(\delta(e)) \le b(e)$ holds by (3.7b) for $\mathrm{IM}(G, b, c)$. If $b'(e, v) = \lceil b(e)/2 \rceil$, then the following holds:

$$x'(\delta(v)) \leq \frac{x(\delta(v))}{2} \leq \frac{b(e)}{2} \leq \left\lceil \frac{b(e)}{2} \right\rceil = a(v).$$

This implies that $x'(\delta(v))$ satisfies (2.7b) in $\mathrm{MA}(G, a, c)$.

Consider the other case, $b'(e, v) < \lceil b(e)/2 \rceil$, i.e., $b(e)$ is odd and $b'(e, v) = \lfloor b(e)/2 \rfloor$. Since $x(\delta(v)) \leq b(e)$ and $a(v) = b'(e, v) = (b(e) - 1)/2$, we have

$$\frac{a(v)}{x(\delta(v))} \geq \frac{b(e) - 1}{2b(e)} = \frac{1}{2} - \frac{1}{2b(e)}.$$

From the assumption, $b(e) \geq \beta_1$ holds, which implies

$$\frac{1}{2} - \frac{1}{2b(e)} \geq \frac{1}{2}\left(1 - \frac{1}{\beta_1}\right).$$

By these inequalities, $x'(\delta(v))$ satisfies (2.7b) for $\mathrm{MA}(G, a, c)$. □

Lemma 3.5. *Let $x \in \mathbb{R}_+^E$ satisfy* (2.7a) *and* (2.7b) *of* $\mathrm{MA}(G, a, c)$*. Then vector*

$$x' = \left(1 - \frac{1}{2\lfloor 3\beta_2/2 \rfloor + 1}\right) x$$

satisfies (2.7c) *for* $\mathrm{MA}(G, a, c)$*, where* $\beta_2 = \min\{\lfloor b(e)/2 \rfloor \mid e \in E, b(e) \neq 0\}$.

Proof. Let U be a non-empty subset of V, and F be a subset of $\delta(U)$ which can be empty. It suffices to show that the following holds:

$$x'(E[U]) + x'(F) \leq \left\lfloor \frac{a(U) + c(F)}{2} \right\rfloor.$$

We can assume that U contains no vertices v such that $a(v) = 0$ (i.e., $x(\delta(v)) = 0$) because the above inequality for such U and any F is obtained from the one for $U - v$ and $F - \delta(v)$.

Since x satisfies (2.7b) for $\mathrm{MA}(G, a, c)$,

$$2x(E[U]) + x(\delta(U)) = \sum_{v \in U} x(\delta(v)) \leq a(U)$$

holds, from which we have

$$x(E[U]) \leq \frac{a(U) - x(\delta(U))}{2}. \tag{3.8}$$

From (2.7a), $x(F) = \sum_{e \in F} x(e) \leq \sum_{e \in F} c(e) = c(F)$ holds. From this inequality and (3.8), we have

$$x(E[U]) + x(F) \leq \frac{a(U) + c(F) - (x(\delta(U)) - x(F))}{2}.$$

Since $x(\delta(U)) - x(F) \geq 0$ holds by $F \subseteq \delta(U)$, we have

$$x(E[U]) + x(F) \leq \frac{d(U) + c(F)}{2}. \tag{3.9}$$

The gap between $(a(U) + c(F))/2$ and $\lfloor (a(U) + c(F))/2 \rfloor$ depends on the parity of $a(U) + c(F)$. Therefore we only have to consider the case where $a(U) + c(F)$ takes an odd value. We consider the following three cases.

Case 1: $|U| = 1$. Let $U = \{v\}$. Then $x(E[U]) = 0$. Therefore the left hand side of (2.7c) equals to $x(F)$. Since $a(U) + c(F) = a(v) + c(F)$ is assumed to be odd, it holds that $a(v) \neq c(F)$, which implies $a(v) + c(F) \geq 2\min\{a(v), c(F)\} + 1$. From (2.7a), $x(F) \leq c(F)$ holds. Moreover, $x(F) \leq x(\delta(v)) \leq a(v)$ holds since $F \subseteq \delta(v)$. Therefore we have

$$x(F) \leq \min\{a(v), c(F)\} \leq \frac{a(U) + c(F) - 1}{2} = \left\lfloor \frac{a(U) + c(F)}{2} \right\rfloor.$$

Case 2: $|U| = 2$. Let $U = \{v_1, v_2\}$, $F_1 = \delta(v_1) \cap F$, and $F_2 = \delta(v_2) \cap F$. Then $a(U) + c(F) = d(v_1) + d(v_2) + c(F_1) + c(F_2)$. From the facts that $\delta(v_1) \cup F_2 \supseteq E[U] \cup F$ and that $\delta(v_2) \cup F_1 \supseteq E[U] \cup F$, we have

$$x(E[U]) + x(F) \leq \min\{x(\delta(v_1)) + x(F_2), x(\delta(v_2)) + x(F_1)\}. \tag{3.10}$$

It holds that $x(\delta(v_1)) \leq a(v_1)$ and $x(\delta(v_2)) \leq a(v_2)$ from (2.7b). Moreover, we have $x(F_1) \leq c(F_1)$ and $x(F_2) \leq c(F_2)$ from (2.7a). These relations and inequality (3.10) lead to

$$x(E[U]) + x(F) \leq \min\{a(v_1) + c(F_2), a(v_2) + c(F_1)\}. \tag{3.11}$$

On the other hand, since $a(U) + c(F)$ is assumed to be odd, it holds that $a(v_1) + c(F_2) \neq a(v_2) + c(F_1)$, which implies that

$$\min\{a(v_1) + c(F_2), a(v_2) + c(F_1)\} \leq \left\lfloor \frac{a(U) + c(F)}{2} \right\rfloor. \tag{3.12}$$

From (3.11) and (3.12), we have (2.7c) for $\mathrm{MA}(G, d, c)$.

Case 3: $|U| \geq 3$. Since $b(e) \geq \min\{b(e) \mid e \in E, b(e) \neq 0\}$ for all $e \in E$, it holds that $a(v) \geq \beta_2$ for all $v \in V$. Hence $a(U) \geq 3\beta_2$. Considering that $a(U) + c(F)$ is odd, we have

$$a(U) + c(F) \geq 2\left\lfloor \frac{3\beta_2}{2} \right\rfloor + 1.$$

From (3.9) and the above inequality,

$$\frac{\lfloor (a(U) + c(F))/2 \rfloor}{x(E[U]) + x(F)} \geq 1 - \frac{1}{a(U) + c(F)} \geq 1 - \frac{1}{2\lfloor 3\beta_2/2 \rfloor + 1}.$$

This completes the proof of the lemma. □

Theorem 3.8. *Let a be a function constructed in algorithm PACK. Then $MA(G, a, c)$ is a polyhedron whose maximum cost extreme points are $f(\beta_1, \beta_2)$-approximate solutions of the (b, c)-induced matching problem for a graph G, where*

$$f(\beta_1, \beta_2) = \frac{1}{2}\left(1 - \frac{1}{\beta_1}\right) \cdot \left(1 - \frac{1}{2\lfloor 3\beta_2/2 \rfloor + 1}\right),$$

$\beta_1 = \min\{b(e) \mid e \in E,\ b(e)$ *is odd*$\}$ *and* $\beta_2 = \min\{\lfloor b(e)/2 \rfloor \mid e \in E, b(e) \neq 0\}$.

Proof. Let $\bar{x} \in \mathbb{R}_+^E$ be a maximum cost extreme point of $\mathrm{MA}(G, a, c)$. Since $\mathrm{MA}(G, a, c)$ is an integer polyhedron, $\bar{x}$ is an integer vector. We have already observed that an integer vector in $\mathrm{MA}(G, a, c)$ is a (b, c)-induced matching. Hence $\bar{x}$ is a (b, c)-induced matching.

In what follows, let us consider the cost of $\bar{x}$. We let OPT denote the maximum cost of (b, c)-induced matchings for G, and x^* denote a vector in $\mathrm{IM}(G, b, c)$ which attains the maximum cost. Since $\mathrm{IM}(G, b, c)$ contains an optimal solution to problem (3.6), we have

$$\mathrm{OPT} \leq w^T x^*.$$

By Lemmas 3.4 and 3.5, we can see that vector $f(\beta_1, \beta_2)x^*$ belongs to $\mathrm{MA}(G, a, c)$. By the maximality of $w^T \bar{x}$ over $\mathrm{MA}(G, a, c)$, it holds that

$$f(\beta_1, \beta_2) w^T x^* \leq w^T \bar{x}.$$

From the above two inequalities, we have

$$f(\beta_1, \beta_2)\mathrm{OPT} \leq w^T \bar{x},$$

as required. □

The above theorem is equivalent to saying that the approximation factor of algorithm PACK is $f(\beta_1, \beta_2)$ because algorithm PACK outputs a maximum cost vector over the polyhedron $\mathrm{MA}(G, a, c)$.

Corollary 3.1. *Let* $\beta_1 = \min\{b(e) \mid e \in E,\ b(e) \text{ is odd}\}$ *and* $\beta_2 = \min\{\lfloor b(e)/2 \rfloor \mid e \in E, b(e) \neq 0\}$. *Then the approximation factor of algorithm PACK is* $f(\beta_1, \beta_2)$. □

Note that $f(\beta_1, \beta_2) = 0$ if E contains an edge e such that $b(e) = 1$. We consider the case where $b(e) = 0$ or $b(e) \geq 2$ for all $e \in E$ (i.e., $\beta_1 \geq 3$ and $\beta_2 \geq 1$). In particular, for $\beta_1 = 3$ and $\beta_2 = 1$, $f(\beta_1, \beta_2) = 2/9$ holds.

Corollary 3.2. *If* $b(e) = 0$ *or* $b(e) \geq 2$ *for all* $e \in E$, *then algorithm PACK achieves an approximation factor of* $2/9$. □

Figure 3.5 shows a tight example for the above analysis in the case of $\beta_1 = 3$ and $\beta_2 = 1$. The example consists of a graph $G = (V_1 \cup V_2, E_1 \cup E_2)$, an edge cost $w : E_1 \cup E_2 \to \mathbb{Q}_+$, and capacities $b, c : E_1 \cup E_2 \to \mathbb{Z}_+$. The vertex set consists of two disjoint sets V_1 and V_2, where $|V_1| = |V_2|$. Edge set E_1 forms a Hamiltonian cycle on V_1. Each edge in E_2 joins a vertex in V_1 and another in V_2 so that $e \cap f = \emptyset$ for each $e, f \in E_2$. Figure 3.5 shows an example, where vertices in V_1 (resp., in V_2) are represented by black circles (resp., white circles) and edges in E_1 (resp., in E_2) are represented by gray lines (resp., black lines). Capacity b is set to be

$$b(e) = \begin{cases} \infty & \text{if } e \in E_1 \\ 3 & \text{if } e \in E_2. \end{cases}$$

Capacity $c(e) = \infty$ for all $e \in E_1 \cup E_2$. If the cost of edges in E_2 is large enough, then the maximum cost over $\mathrm{IM}(G, b, c)$ is achieved by

$$x^*(e) = \begin{cases} \frac{3}{2} & \text{for } e \in E_1 \\ 0 & \text{for } e \in E_2. \end{cases}$$

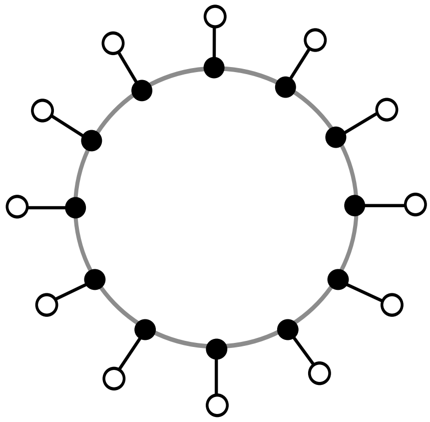


Figure 3.5: A tight example for the analysis in Corollary 3.2

Algorithm PACK may compute $a(v) = 1$ for $v \in V_1$ and $a(v) = 2$ for $v \in V_2$. For the resulting instance (G, a, c) of (a, c)-matching problem, we need to multiply x^* by $2/9$ in order to satisfy (2.7c) of $\mathrm{MA}(G, a, c)$ for $U = V_1$ and $F = \phi$.

3.4 Hyperedge dominating set problem

In this section we discuss the hypergraph version of the edge dominating set problem. For a hypergraph $H = (V, E)$, a *hyperedge dominating set* (HEDS) $F \subseteq E$ is defined as a set of edges dominating all hyperedges in E, i.e., each hyperedge $e \in E$ is contained in F or shares at least one vertex with an edge in F. Then the problem is the following.

> **Hyperedge dominating set (HEDS) problem**
> Given a hypergraph $H = (V, E)$ and a hyperedge cost $w : E \to \mathbb{Q}$, find a 3 cost hyperedge dominating set.

This problem is formulated as the following integer programming.

$$\begin{array}{lll} \text{minimize} & w^T x & \\ \text{subject to} & x(\delta(e)) \geq 1 & \text{for each } e \in E, \\ & x \in \mathbb{Z}_+^E. & \end{array} \tag{3.13}$$

Let $\mathrm{HEDS}(H)$ denote the feasible region obtained by relaxing the integrality constraints in (3.13) into $x \geq 0$. To obtain an approximate solution to the HEDS problem, we transform a given instance of the HEDS problem to an instance of the set cover problem, which we defined in Section 2.4.2. Recall that the set cover problem can be considered as a hypergraph version of the edge cover problem.

Since the HEDS problem is a special case of the set cover problem, the HEDS problem can be reduced to the set cover problem directly. Let $(H = (V, E), w)$ be a given instance of the HEDS problem. Construct a hypergraph $H' = (V', E')$ such that its vertex set V' consists of vertices v'_e corresponding to its edges $e \in E$ and edge set E' consists of $e'_e = \{v'_{e''} \mid e'' \in \delta(e)\}$

corresponding to $\delta(e)$. A component of the cost vector $w'(e'_e)$ is set to be $w(e)$. Then, it is easy to see that a set cover for (H', w') gives an HEDS for (H, w) of same cost and vice versa. Let d be the maximum size of a hyperedge in E', i.e., the maximum size of $\delta(e)$ for all $e \in E$, where $d = \mathrm{O}(|V|^k)$ holds for the maximum size k of a hyperedge in H. By Theorem 2.6, this direct reduction gives a θ_d-approximation algorithm for the HEDS problem. Note that $\theta_d = \mathrm{O}(k \log |V|)$.

In our algorithm, an instance of the HEDS problem is transformed into an instance of the set cover problem.

Algorithm HYPER

Input: A hypergraph $H = (V, E)$ and an edge cost $w : E \to \mathbb{Q}_+$

Output: An HEDS for H

1: Compute a vector $x^* \in \mathrm{HEDS}(H)$ minimizing $w^T x^*$;
2: $V' := \{v \in V \mid x^*(\delta(v)) = \max_{u \in e} x^*(\delta(u))$ for some $e \in E\}$;
3: $E' := \{e \cap V' \mid e \in E\}$; $w' := w_{E'} \in \mathbb{Q}_+^{E'}$;
4: Compute a set cover $\bar{x}$ by algorithm SETCOVER to an instance with $H_{x^*} = (V', E')$ and w';
5: Output $\bar{x}_E$ as an HEDS for H;

To prove that the approximation factor of this algorithm is $k\theta_k$ by using Theorem 2.6, we show that the vector kx^* is contained in $\mathrm{SC}(V', E')$.

Lemma 3.6. *Let $x \in \mathrm{HEDS}(H)$ for a hypergraph $H = (V, E)$ and $H_x = (V', E')$ be the hypergraph obtained in algorithm HYPER from x. Then, for $k = \max_{e \in E} |e|$, the vector $kx_{E'} \in \mathbb{R}^{E'}$ is contained in* $\mathrm{SC}(V', E')$.

Proof. Suppose that $v \in V'$ is a vertex in a hyperedge $e \in E'$ such that $x(\delta(v)) \geq x(\delta(u))$ for all $u \in e$. Since $\sum_{u \in e} x(\delta(u)) \geq x(\delta(e)) \geq 1$, we have $x(\delta(v)) \geq 1/k$. Therefore $kx \langle E' \rangle (\delta(v)) \geq 1$, which means that $kx_{E'} \in \mathrm{SC}(V', E')$. □

Theorem 3.9. *Algorithm HYPER achieves an approximation factor of $k\theta_k$ for the HEDS problem, where k is the maximum size of hyperedges.*

Proof. Let $\tilde{x} \in \mathrm{SC}(V', E')$ be a vector minimizing $w'^T \tilde{x}$. Then by Theorem 2.6, $w'^T \bar{x} \leq \theta_k w'^T \tilde{x}$ follows. In addition, Lemma 3.6 implies that $w'^T \tilde{x} \leq k w'^T x^*_{E'}$. Hence

$$w'^T \bar{x} \leq \theta_k w'^T \tilde{x} \leq k\theta_k w'^T x^*_{E'} = k\theta_k w^T x^*.$$

Since $w'^T \bar{x}$ is the cost of solution algorithm HYPER outputs and $w^T x^*$ is a lower bound of the optimal cost, it completes the proof. □

Note that the approximation factor $k\theta_k = \mathrm{O}(k \log k)$ of algorithm HYPER is superior to that of the algorithm obtained from the direct reduction if $k\theta_k < \theta_d$, i.e., if H is a dense hypergraph such that $d = \Omega(|V|^k)$.

3.5 Generalized (a, c)-edge cover

In the (b, c)-EDS problem, the constraint $x(\delta(e)) \geq b(e)$ is considered for every $e \in E$. However, we can see that algorithm DOMINATE(f) in Section 3.2 can deal with this constraint also for a pair of vertices that are joined by no edge in E, i.e., $x(\delta(u) \cup \delta(v)) \geq b(uv)$ for any pair $\{u, v\}$ of vertices. Motivated by this observation, we consider the following problem.

> **Generalized (a, c)-edge cover problem**
> Given an undirected graph $G = (V, E)$, an edge cost $w : E \to \mathbb{Q}_+$, a family $\mathcal{V} \subseteq 2^V$ of subsets of V, a demand $a : \mathcal{V} \to \mathbb{Z}_+$ and capacity $c : E \to \mathbb{Z}_+$, find a minimum cost multiset F of edges such that $\sum_{v \in U} |\delta(v; F)| \geq a(U)$ for each $U \in \mathcal{V}$ and F contains at most $c(e)$ copies of each $e \in E$.

We call feasible solutions for this problem *generalized (a, c)-edge covers*. An integer programming formulation of the problem is given as follows.

$$
\begin{array}{lll}
\text{minimize} & w^T x & \\
\text{subject to} & \sum_{v \in U} x(\delta(v)) \geq a(U) & \text{for each } U \in \mathcal{V}, \\
& x(e) \leq c(e) & \text{for each } e \in E, \\
& x \in \mathbb{Z}_+^E. &
\end{array} \tag{3.14}
$$

Note that if $\mathcal{V} = \{\{v\} \mid v \in V\}$, then problem (3.14) is equivalent to the (a, c)-edge cover problem (2.4). If $\mathcal{V} = \{\{u, v\} \mid (u, v) \in E\}$, then problem (3.14) seems similar to the (b, c)-EDS problem, but its first constraint $x(\delta(u)) + x(\delta(v)) \geq a(e)$ on each $e = (u, v) \in E$ is different from the constraint $x(\delta(e)) \geq a(e)$ for the (b, c)-EDS. Let $\mathrm{DC}(G, \mathcal{V}, a, c)$ denote the set of all vectors $x \in \mathbb{R}_+^E$ satisfying the inequalities in (3.14), i.e., the relaxation of the covering problem. We show that problem (3.14) is approximable by algorithm COVER(f) described below.

Algorithm COVER(f)

Input: A simple undirected graph $G = (V, E)$, an edge cost $w : E \to \mathbb{Q}_+$, a family $\mathcal{V} \subseteq 2^V$ of subsets of V, a demand $a : \mathcal{V} \to \mathbb{Z}_+$, a capacity $c : E \to \mathbb{Z}_+$, and a real $f > 0$

Output: A generalized (a, c)-edge cover or "INFEASIBLE"

if $\mathrm{DC}(G, \mathcal{V}, a, c) = \emptyset$ **then**
  Output "INFEASIBLE" and halt
end if;
$E' := \emptyset$;
Compute $x^* \in \mathrm{DC}(G, \mathcal{V}, a, c)$ minimizing $w^T x^*$;
for each $e \in E$ with $f x^*(e) > c(e)$ **do**
  $\bar{x}(e) := c(e)$; $E' := E' \cup \{e\}$;
  for each $U \in \mathcal{V}$ with $e \in E[U]$ **do**
    $a(U) := \max\{0, a(U) - 2c(e)\}$
  end for;
  for each $U \in \mathcal{V}$ with $e \in \delta(U)$ **do**

    $a(U) := \max\{0, d(U) - c(e)\}$
  end for
end for;
for each $U \in \mathcal{V}$ **do**
  if $x^*(\delta(v) - E') \geq x^*(\delta(u) - E')$ for all $u \in U$ **then**
    $a'_{x^*}(v, U) := a(U)$
  else
    $a'_{x^*}(v, U) := 0$
  end if
end for;
for each $v \in V$ **do**
  $\tilde{a}_{x^*}(v) := \max_{U \in \mathcal{V}: v \in U} a'_{x^*}(v, U)$
end for;
Compute a minimum cost $(\tilde{a}_{x^*}, c)$-edge cover $\bar{x}_{E-E'}$ for $G' = (V, E - E')$ and $w_{E-E'}$;
Output $\bar{x}$ as a generalized (a, c)-edge cover.

For each $U \in \mathcal{V}$, the vertex $v = \arg\max_{u \in U} x^*(\delta(u) - E')$ satisfies

$$\sum_{u \in U} \bar{x}(\delta(u) - E') \geq \bar{x}(\delta(v) - E') \geq \tilde{a}_{x^*}(v) \geq a(U) - 2c(E[U] \cap E') - c(\delta(U) \cap E').$$

Since

$$\sum_{u \in U} \bar{x}(\delta(u)) \geq \sum_{u \in U} \bar{x}(\delta(u) - E') + 2\bar{x}(E[U] \cap E') + \bar{x}(\delta(U) \cap E') \geq a(U),$$

we can see that $\bar{x}$ is a generalized (a, c)-edge cover. In which follows, we discuss the approximation factor of COVER(f). It can be derived analogously to that of algorithm DOMINATE(f). First, let us consider the case of $c(e) = +\infty$. Notice that $E' = \emptyset$ for any f in this case.

Lemma 3.7. *Let $x \in \mathrm{DC}(G, \mathcal{V}, a, +\infty)$ and $k = \max_{U \in \mathcal{V}} |U|$. The vector kx satisfies* (2.5a) *and* (2.5b) *of* $\mathrm{EC}(G, \tilde{a}_x, +\infty)$*, where $\tilde{a}_x : V \to \mathbb{Z}_+$ is the function obtained from x in COVER(f).*

Proof. Since $x \in \mathbb{R}_+^{E-E'}$, vector kx satisfies (2.5a) for $\mathrm{EC}(G, \tilde{a}_x, +\infty)$. We show that kx satisfies (2.5b) as well, i.e., $kx(\delta(v)) \geq \tilde{a}_x(v)$ for each $v \in V$. Let v be a vertex in V. If $\tilde{a}_x(v) = 0$, then $kx(\delta(v)) \geq 0 = \tilde{a}_x(v)$ holds. Now assume $\tilde{a}_x(v) > 0$. There exists a subset $U \in \mathcal{V}$ such that $x(\delta(v)) \geq x(\delta(u))$ holds for all $u \in U$ and $\tilde{a}_x(v) = a'_x(v, U) = a(U)$. From this inequality and the condition $\sum_{u \in U} x(\delta(u)) \geq a(U)$ for $\mathrm{DC}(G, \mathcal{V}, a, +\infty)$, we have

$$kx(\delta(v)) \geq |U| x(\delta(v)) \geq \sum_{u \in U} x(\delta(u)) \geq a(U) = a'_x(v, U) = \tilde{a}_x(v).$$

This implies that kx satisfies (2.5b) for $\mathrm{EC}(G', \tilde{a}_x, +\infty)$. □

Lemmas 3.2 and 3.7 now imply the following theorem.

Theorem 3.10. *Algorithm COVER(f) achieves an approximation factor of*

$$k \cdot \left(1 + \frac{1}{2\lfloor 3\beta/2 \rfloor + 1}\right) \left(\leq \frac{4}{3} k\right)$$

for the generalized $(a, +\infty)$-edge cover problem, where $k = \max\{|U| \mid U \in \mathcal{V}\}$ and $\beta = \min_{U \in \mathcal{V}, a(U) \neq 0} a(U)$.

Proof. Let $y = k \cdot (1 + 1/(2\lfloor 3\beta/2 \rfloor + 1))\, x^*$. By Lemmas 3.2 and 3.7, it holds that $y \in \mathrm{EC}(G, \tilde{a}_x, +\infty)$, which implies that $w^T y$ is at least the minimum cost over $\mathrm{EC}(G, \tilde{a}_x, +\infty)$. Since the algorithm outputs a vector of minimum cost over $\mathrm{EC}(G, \tilde{a}_x, +\infty)$ and $w^T x^*$ is a lower bound of the optimal cost, the proof is completed. □

For the case where $c(e)$ is finite, we can derive an approximation factor similarly for the (b, c)-EDS problem, if the parameter f is set appropriately. Particularly, COVER$(4k/3)$ achieves the factor of $4k/3$.

Theorem 3.11. *Algorithm COVER$(4k/3)$ achieves an approximation factor of $4k/3$ for the generalized (a, c)-edge cover problem, where $k = \max\{|U| \mid U \in \mathcal{V}\}$.* □

Chapter 4

Splitting and Detachment with Local Edge-Connectivity

In Section 2.5, we defined graph transformations, called the splitting and the detachment, and reviewed prior results on them. This chapter gives several new results on those transformations preserving local edge-connectivity.

4.1 Strongly splittable pair

For an undirected graph $G = (V, E)$ and a vertex $s \in V$, let

$$r_G(x, y) = \begin{cases} \lambda(x, y; G) & \text{if } x, y \in V - s, \\ \min\{\deg(s; G) - 2, \lambda(x, y; G)\} & \text{if } s \in \{x, y\}. \end{cases}$$

Obviously G is r_G-edge-connected. We call a pair $\{e, f\}$ of edges incident to s *strongly splittable* if $G^{e,f}$ is also r_G-edge-connected, i.e., splitting such a pair preserves the local edge-connectivity between every two $x, y \in V$–s and that between s and the others up to $\deg(s; G)-2$. Clearly a strongly splittable pair is also splittable.

Define $f_{r_G}(X) = \max_{u \in X, v \in V-X} r_G(u, v)$ and $h(X) = d(X; G) - f_{r_G}(X)$ for each $X \subseteq V - s$. Notice that $h(X) \geq 0$ holds for every $X \subseteq V - s$. We call $X \subseteq V - s$ *dangerous* if $h(X) \leq 1$. Strongly splittable pairs can be characterized by the existence of dangerous sets.

Theorem 4.1. *A pair $\{e = us, f = vs\}$ of edges incident to s is strongly splittable if and only if no dangerous set $X \subseteq V - s$ contains both u and v.*

Proof. Suppose that a dangerous set $X \subseteq V - s$ contains both u and v. Then $d(X; G^{e,f}) = d(X; G) - 2 \leq f_{r_G}(X) - 1$ holds. This indicates that $\{e, f\}$ is not strongly splittable. Hence necessity holds.

On the other hand, assume that $\{e, f\}$ is strongly splittable. Then $d(X; G^{e,f}) \geq f_{r_G}(X)$ holds for every $X \subseteq V - s$ that contains both u and v. Since $d(X; G) = d(X; G^{e,f}) + 2$, it holds that $d(X; G) \geq f_{r_G}(X) + 2$. Hence X is not dangerous. Accordingly, sufficiency holds. □

In the following theorem, we give a condition for a graph to have a strongly splittable pair.

Theorem 4.2. *Let $G = (V, E)$ be an undirected connected graph, and $s \in V$ be a vertex with* $\deg(s) \neq 3$. *If no cut-edge is incident to s, then there exists at least one strongly splittable pair $\{e = us, f = vs\}$ of edges, where e and f can be chosen so that $u \neq v$ unless s is adjacent to only one vertex. No new cut-edge will be created after splitting $\{e, f\}$.*

Proof. Let us consider a new graph $G' = (V + s', E \cup E')$ where E' consists of $\deg(s; G) - 2$ edges between s and s'. Then it hold $\lambda(x, y; G') = r_G(x, y)$ for any $x, y \in V - s$, $\lambda(s', y; G') = r_G(s, y)$ for any $y \in V - s$ and $\deg(s; G') = 2\deg(s; G) - 2$. By Theorem 2.13, edges incident to s can be partitioned into $\deg(s; G) - 1$ disjoint pairs such that splitting them preserves the local edge-connectivity between vertices in $V - s + s'$. Notice that at least one of those pairs consists of only edges in E. This is exactly a strongly splittable pair.

If a pair of two parallel edges us is splittable, there is no set X such that $u \in X$, $s \notin X$, $d(X; G) - \max_{x \in X, y \in V - X} r_G(x, y) \leq 1$ by Lemma 2.3. This implies that any pair of us and another arbitrary edge is also splittable. From this fact and the above existence of a strongly splittable pair, the existence of a strongly splittable pair $\{us, vs\}$ with $u \neq v$ follows when the number of neighbors of s is more than 1.

Finally we show that the splitting by a strongly splittable pair does not generate a new cut-edge. For a strongly splittable pair $\{e = us, f = vs\}$ of edges in G, assume that $G^{e,f}$ contains a new cut-edge $e' = zw$. If $e' = zw$ is an existing edge in G, then $1 = \lambda(z, w; G^{e,f}) = r_G(z, w)$, implying that $\lambda(z, w; G) = 1$ by the definition of $r_G(z, w)$, contradicting that zw was not a cut-edge in G. Next assume that $e' = zw$ is a new edge in $G^{e,f}$, i.e., $zw = uv$. In this case, $G^{e,f} - e'$ is not connected and has a component not containing s, implying that s and this component was joined by a cut-edge, a contradiction to the assumption. $\square$

Theorem 4.2 implies the existence of strongly splittable pairs unless a cut-edge is incident to s or $\deg(s; G) = 3$. This has a close relationship to the parsimonious property of the Steiner network problem introduced in Section 2.3. In fact, M. X. Goemans and D. J. Bertsimas [30] proved this property by showing that every Eulerian graph admits a strongly splittable pair, which is a weaker version of Theorem 4.2. From Theorem 4.2, we can derive an integer programming version of the parsimonious property for the Steiner network problem.

Corollary 4.1. *If an edge cost $w : \binom{V}{2} \to \mathbb{Q}_+$ is metric, then an instance (V, w, r) of the Steiner network problem without edge capacity has an optimal solution G such that* $\deg(v; G)$ *is* $\max_{u \in V - v} r(u, v)$ *or* $\max_{u \in V - v} r(u, v) + 1$.

Proof. Suppose $\deg(v; G) > \max_{u \in V - v} r(u, v) + 1$ for a vertex $v \in V$. By Theorem 4.2, we can obtain another r-edge connected graph G' with $\deg(v; G') = \deg(v; G) - 2 \geq \max_{u \in V - v} r(u, v)$ by splitting an appropriate pair of edges incident to v. Since w is metric, cost of G' is at most that of G. Hence, by repeating this operation, we can obtain another optimal solution such that the degree of each vertex v is $\max_{u \in V - v} r(u, v)$ or $\max_{u \in V - v} r(u, v) + 1$. $\square$

Theorem 4.2 provides several corollaries on detachments preserving local edge-connectivity.

Corollary 4.2. *For a vertex $s \in V$ with $\deg(s;G) \geq 4$ which has no incident cut-edge in a connected graph $G = (V,E)$, and degree specification $g(s) = (V_s = \{s_1, s_2\}, \{\rho(s_1) = d(s;G) - 2, \rho(s_2) = 2\}$, let*

$$r(x,y) = \begin{cases} \lambda(x,y;G) & \text{if } \{x,y\} \cap \{s_1,s_2\} = \emptyset, \\ \min\{\lambda(s,z;G), \rho(s_i)\} & \text{if } \{s_i\} = \{x,y\} \cap \{s_1,s_2\} \text{ and } \{z\} = \{x,y\} - \{s_1,s_2\}, \\ 2 & \text{if } \{x,y\} = \{s_1,s_2\}. \end{cases}$$

Then there exists an r-edge-connected $g(s)$-detachment of G. Two edges incident to s_2 in it are not parallel unless s is adjacent to only one vertex in G

Proof. By Theorem 4.2, G has a strongly splittable pair $\{e = us, f = vs\}$. That is, $\lambda(x,y;G^{e,f}) \geq r_G(x,y) = \lambda(x,y;G)$ holds for all pairs $x,y \in V - s$, and $\lambda(s,y;G^{e,f}) \geq r_G(s,y) = \min\{\deg(s;G) - 2, \lambda(s,y;G)\}$ holds for all $y \in V - s$. Let G' be a graph obtained from $G^{e,f}$ by regarding s as s_1 and by replacing edge uv with two edges us_2 and vs_2 introducing a new vertex s_2. Observe that $\lambda(x,y;G') = \lambda(x,y;G^{e,f}) \geq r(x,y)$ for vertices x,y with $\{x,y\} \cap \{s_2\} = \emptyset$. We first show that $\lambda(s_1,s_2;G') \geq 2$. Assume $\lambda(s_1,s_2;G') \leq 1$; there is a minimal subset X with $s_2 \in X \subseteq (V-s) \cup \{s_2\}$ and $d(X;G') \leq 1$. By the minimality, X induces a connected component. Suppose that h is an edge whose one end vertex is in X and the other is in $V \cup \{s_2\} - (X \cup \{s\})$, i.e., $d(X;G') = 1$. Then removing h disconnects X from the other vertices in G' but does not do so in G. This implies that h is a new cut-edge, a contradiction. If $d(X;G') = 0$, then $\lambda(s_1,v;G') = \lambda(s_1,v;G^{e,f}) = 0$ for each $v \in X - s_2$, which contradicts the strongly splittability of $\{e,f\}$. Hence $\lambda(s_1,s_2;G') \geq 2$ holds. Finally we show that $\lambda(s_2,y;G') \geq r(s_2,y) = \min\{\lambda(s,y;G), \rho(s_2) = 2\}$ holds. Assume $\lambda(s_2,y;G') < \min\{\lambda(s,y;G), \rho(s_2) = 2\}$. Then $\lambda(s_2,y;G') \leq 1$. By $\lambda(s_1,s_2;G') \geq 2$, there is a subset $Y \in V - s$ with $y \in Y$ and $d(Y;G') = \lambda(s_2,y;G') \leq 1$. This also implies that $\lambda(s_1,y;G') \leq d(Y;G') = \lambda(s_2,y;G') < \min\{\lambda(s,y;G), \rho(s_2) = 2\}$, which is a contradiction to the strongly splittability of $\{e,f\}$. Therefore G' is a desired $g(s)$-detachment of G. □

The next corollary is immediate from Corollary 4.2.

Corollary 4.3. *For a vertex $s \in V$ which has no incident cut-edge in a connected graph $G = (V,E)$, and a degree specification $g(s) = (V_s = \{s_1, \ldots, s_p\}, \rho)$ with $\rho(s_1) = \deg(s;G) - 2(p-1) \geq 2$ and $\rho(s_i) = 2$, $s_i \in V_s - s_1$, let*

$$r(x,y) = \begin{cases} \lambda(x,y;G) & \text{if } \{x,y\} \cap V_s = \emptyset, \\ \min\{\lambda(s,z;G), \rho(s_i)\} & \text{if } \{s_i\} = \{x,y\} \cap V_s \text{ and } \{z\} = \{x,y\} - V_s, \\ 2 & \text{if } \{x,y\} \subseteq V_s. \end{cases}$$

Then there exists an r-edge-connected $g(s)$-detachment of G. □

Furthermore, we can extend the above result to global detachments.

Corollary 4.4. *For a graph $G = (V,E)$, let g be a degree specification such that $\rho(v_1) = \deg(v;G) - 2(p_v - 1) \geq 2$ and $\rho(v_i) = 2$ $(i = 2, \ldots, p_v)$ hold for $V_v = \{v_1, \ldots, v_{p_v}\}$, where $|V_v| = 1$ if $\deg(v;G) \leq 3$ or a cut-edge is incident to v. Let*

$$r(x,y) = \begin{cases} \min\{\lambda(u,v;G), \rho(x), \rho(y)\} & \text{if } x \in V_u,\ y \in V_v \text{ with } u \neq v, \\ 2 & \text{otherwise.} \end{cases}$$

Then there exists an r-edge-connected g-detachment of G. Furthermore, if $|V| \geq 3$ and $\rho(v_1)$ is uniform for all $v \in V$, then two edges incident to $v_i \in V_v - \{v_1\}$ are not parallel for each $v \in V$.

Proof. Adopting Corollary 4.3 consecutively for each vertex in G gives a required r-edge-connected g-detachment of G. Hence let us consider the second argument about the case in which $|V| \geq 3$ and $\rho(v_1) = \rho$ for all $v \in V$. Let $s \in V$ be a vertex with $\deg(s; G) > \rho$. If s is adjacent to only one vertex (say w), then w has another neighbor in $V - s$. Hence $\deg(w; G) > \deg(s; G) > \rho$. This implies that if G has a vertex with degree of more than ρ, G also has a vertex such that the number of its neighbors is at least 2. Hence by repeating to separate such a vertex, we can obtain an r-edge-connected g-detachment of G □

4.2 Strongly splittable pair containing a specified arc or edge

For splittable pairs, it was shown in Theorem 2.13 that there exists a splittable pair containing a specified edge incident to s if no cut-edge is incident to s and $\deg(s; G)$ is even. It is a natural question to ask whether the similar argument holds for strongly splittable pairs. Unfortunately we have a counterexample to this, as shown in Figure 4.1. In this section, we will prove that the answer to the question is affirmative for Eulerian digraphs and undirected graphs.

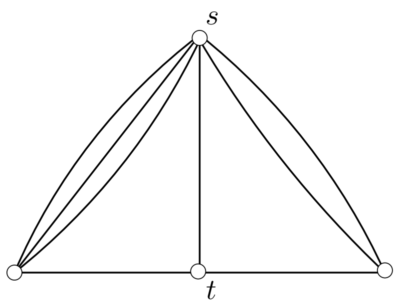


Figure 4.1: A graph that has no strongly splittable pair at s containing edge st

First, let us consider digraphs. Let $D = (V, A)$ be a digraph, and $s \in V$ be a specified vertex. *Strong splittability* for digraph D is defined in the same way with undirected graphs except for that r_G is replaced by a function

$$r_D(x, y) = \begin{cases} \lambda(x, y; D) & \text{if } x, y \in V - s, \\ \min\{\deg^+(s; D) - 1, \lambda(x, y; D)\} & \text{if } x = s, \\ \min\{\deg^-(s; D) - 1, \lambda(x, y; D)\} & \text{if } y = s. \end{cases}$$

That is to say, splitting a strongly splittable pair preserves the local edge-connectivity from s to the other vertices up to $\deg^+(s; D) - 1$, and from the other vertices to s up to $\deg^-(s; D) - 1$, in addition to that between every two vertices in $V - s$. Note that D is r_D-edge-connected.

Now we assume that D is Eulerian unless stated otherwise. Then $d^+(X; D) = d^-(X; D)$ holds for any non-empty $X \subset V$. Hence we denote the value of $d^+(X; D) = d^-(X; D)$ by $d(X; D)$ in the following. Notice that $r_D(x, y) = r_D(y, x)$ also holds in this case for every

$x, y \in V$. In addition, we also assume that D has no loop incident to a designated vertex s. We can easily see that any pair containing a loop is strongly splittable.

For a non-empty set $X \subseteq V - s$, let

$$f_{r_D}(X) = \max_{x \in X, y \in V - X} r_D(x, y)$$

and

$$h(X) = d(X; D) - f_{r_D}(X).$$

Since D is r_D-edge-connected, $d(X; D) \geq f_{r_D}(X)$ holds for all non-empty and proper subsets X of V, and hence $h(X) \geq 0$ for $\emptyset \neq X \subset V$. A subset X of vertices is called *tight* if $h(X) = 0$ and $\emptyset \neq X \subseteq V - s$. Notice that no tight subset X is assumed to contain s. Tight sets give a characterization of strongly splittable pairs in Eulerian digraphs.

Lemma 4.1. *A pair $\{e = us, f = sv\}$ of arcs in an Eulerian digraph D is strongly splittable if and only if no tight set contains both of u and v.*

Proof. Let $X \subseteq V - s$ be a tight set (i.e., $h(X) = d(X; D) - f_{r_D}(X) = 0$) containing u and v. Then it holds $d(X; D^{e,f}) = d(X; D) - 1 < R(X) = r_D(x, y)$ for some $x \in X$ and $y \in V - X$, which implies that $\{e, f\}$ is not strongly splittable. Hence the necessity follows.

To show sufficiency, suppose that $\{e = us, f = sv\}$ is not strongly splittable. Then there is a pair $\{x, y\}$ of vertices such that $\lambda(x, y; D^{e,f}) < r_D(x, y)$, which implies that there is a subset X such that $d(X; D^{e,f}) < f_{r_D}(X)$ and $|\{x, y\} \cap X| = 1$. We can assume without loss of generality that $\{x, y, s\} \cap X = \{x\}$. Since D is r_D-edge-connected, $d(X; D) \geq f_{r_D}(X)$ holds. If X contains at most one of u and v, then it holds $f_{r_D}(X) > d(X; D^{e,f}) = d(X; D) \geq f_{r_D}(X)$, a contradiction. Hence X contains both of u and v, and hence $d(X; D) = d(X; D^{e,f}) + 1$ holds. Then it holds $d(X; D) = d(X; D^{e,f}) + 1 < f_{r_D}(X) + 1$, which implies that X is tight, as required. $\square$

Lemma 4.1 deals with only Eulerian digraphs. However, we remark that the statement remains valid for any digraphs provided that a tight set is redefined as a vertex set $X \subseteq V - s$ with $h^+(X) = 0$ or $h^-(X) = 0$, where $h^+(X) = \max_{x \in X, y \in V - X} r_D(x, y) - d^+(X; D)$ and $h^-(X) = \max_{x \in V - X, y \in X} r_D(x, y) - d^-(X; D)$.

We observe the following property of h.

Lemma 4.2. *For any $X, Y \subseteq V - s$, it holds either*

$$\begin{aligned} h(X) + h(Y) \geq \;& h(X \cap Y) + h(X \cup Y) \\ & + d(X - Y, Y - X; D) + d(Y - X, X - Y; D) \end{aligned} \tag{4.1}$$

or

$$\begin{aligned} h(X) + h(Y) \geq \;& h(X - Y) + h(Y - X) \\ & + d(X \cap Y, V - X \cup Y; D) + d(V - X \cup Y, X \cap Y; D). \end{aligned} \tag{4.2}$$

Proof. In Theorems 2.2 and 2.3, we have already seen that

$$\begin{aligned} d(X; D) + d(Y; D) = \;& d(X \cap Y; D) + d(X \cup Y; D) \\ & + d(X - Y, Y - X; D) + d(Y - X, X - Y; D) \end{aligned} \tag{2.9}$$

and

$$\begin{aligned} d(X;D)+d(Y;D) = \;& d(X-Y;D)+d(Y-X;D) \\ & +d(X\cap Y, V-X\cup Y;D)+d(V-X\cup Y, X\cap Y;D) \end{aligned} \tag{2.11}$$

hold. On the other hand, f_{r_D} is weakly supermodular by Theorem 2.5, i.e.,

$$f_{r_D}(X)+f_{r_D}(Y) \leq f_{r_D}(X\cap Y)+f_{r_D}(X\cup Y) \tag{4.5}$$

or

$$f_{r_D}(X)+f_{r_D}(Y) \leq f_{r_D}(X-Y)+f_{r_D}(Y-X). \tag{4.6}$$

If (4.5) holds, we obtain (4.1) by subtracting (4.5) from (2.9). If (4.6) holds, we obtain (4.2) by subtracting (4.6) from (2.11), as required. □

From the above facts, we have the next result on the existence of strongly splittable pairs in Eulerian digraphs, corresponding to Theorem 2.13.

Theorem 4.3. *For an Eulerian digraph $D=(V,A)$, a vertex $s\in V$ and an arc e entering* (*resp., leaving*) *s, there is another arc f leaving s* (*resp., entering s*) *such that $\{e,f\}$ is a strongly splittable pair at s.*

Proof. Let $e=us$. Suppose that there is no strongly splittable pair at s containing e. By Lemma 4.1, there is a tight set X_v for each $v\in\Gamma^+(s)$ which contains both u and v,

Let $v,w\in\Gamma^+(s)$. Then it holds $d(X_v\cap X_w, V-(X_v\cup X_w);D)\geq d(u,s;D)\geq 1$. We see that (4.2) does not hold for X_v and X_w, since otherwise we would have

$$\begin{aligned} 0+0 &= h(X_v)+h(X_w) \\ &\geq h(X_v-X_w)+h(X_w-X_v)+d(X_v\cap X_w, V-(X_v\cup X_w);D) \\ &\quad +d(V-(X_v\cup X_w), X_v\cap X_w;D) \\ &\geq 0+0+1+0, \end{aligned}$$

a contradiction. Therefore by Lemma 4.2, (4.1) holds, i.e.,

$$\begin{aligned} 0+0 \geq h(X_v)+h(X_w) \geq\;& h(X_v\cup X_w)+h(X_v\cap X_w) \\ & + d(X_v-X_w, X_w-X_v)+d(X_w-X_v, X_v-X_w). \end{aligned}$$

This inequality implies that $X_v\cup X_w$ is a tight set in D. From this, we can see that a maximal tight set X contains $\Gamma^+(s)\cup\{u\}$ and satisfies $d(X;D)\geq(s;D)$.

Let $f_{r_D}(X)=r_D(x,y)$, where $x\in X$ and $y\in V-X$. If $y=s$, it holds

$$d(X;D)\geq d(s;D)=\deg^-(s;D)\geq r_D(x,s)+1=r_D(x,y)+1=f_{r_D}(X)+1,$$

where we used the fact that no loop is incident to s, which is assumed above. This implies $h(X)\geq 1$, contradicting tightness of X. Otherwise (i.e., $y\neq s$), it holds $\lambda(x,y;D)=\lambda(x,y;D-s)$ by $\Gamma^+(s)\subseteq X$. We also have $\lambda(x,y;D-s)\leq d(X;D)-d(s;D)$. Hence,

$$f_{r_D}(X)=r_D(x,y)=\lambda(x,y;D)\leq d(X;D)-d(s;D)\leq d(X;D)-1,$$

which implies that $h(X)\geq 1$, a contradiction again. □

We use the following property in Section 4.3.

Theorem 4.4. *For an Eulerian digraph D, a strongly splittable pair $\{e = us, f = sv\}$ can be chosen so that $u \neq v$ unless $|\Gamma^+(s) \cup \Gamma^-(s)| = 1$.*

Proof. By Theorem 4.3, D has a strongly splittable pair. If such a pair consists of arcs us and su, then there is no tight set containing u by Lemma 4.1. Since $|\Gamma^+(s) \cup \Gamma^-(s)| \neq 1$, there is a vertex $v \neq u$ such that $v \in \Gamma^+(s) \cup \Gamma^-(s)$. Assume $v \in \Gamma^+(s)$ without loss of generality. Then $\{us, sv\}$ is strongly splittable in D. □

From Theorem 4.3, we can easily obtain a counterpart for undirected graphs.

Theorem 4.5. *Let $G = (V, E)$ be an Eulerian undirected graph and s be a specified vertex in V. For each edge $e = us \in E$, there is an edge $f = vs$ incident to s such that $\{e, f\}$ is a strongly splittable pair.*

Proof. Since G is Eulerian, we have an orientation $D = (V, A)$ of G such that D is an Eulerian digraph, which satisfies

$$2\lambda(x, y; D) = \lambda(x, y; G) \quad \text{for each } x, y \in V. \tag{4.7}$$

Let e' be the arc in A corresponding to e. By Theorem 4.3, there is another arc f' such that $\{e', f'\}$ is a strongly splittable pair at s in D, i.e., it holds $\lambda(x, y; D^{e',f'}) \geq r_D(x, y)$ for every $x, y \in V$. Let f be the edge in E corresponding to f'. Since $D^{e',f'}$ is also Eulerian, $2\lambda(x, y; D^{e',f'}) = \lambda(x, y; G^{e,f})$ holds for every $x, y \in V$. Notice that $2r_D(x, y) = r_G(x, y)$ also holds for every $x, y \in V$ by (4.7) and $2\deg(s; D) = \deg(s; G)$. Hence it holds

$$\lambda(x, y; G^{e,f}) = 2\lambda(x, y; D^{e',f'}) \geq 2r_D(x, y) = r_G(x, y)$$

for every $x, y \in V$, which implies that $\{e, f\}$ is strongly splittable in G. □

The following theorem gives a counterpart of Theorem 4.4 in undirected graphs.

Theorem 4.6. *For an Eulerian undirected graph G, a strongly splittable pair $\{e = us, f = sv\}$ can be chosen so that $u \neq v$ unless $|\Gamma(s)| = 1$.*

Proof. Let us consider an orientation D of G, which appeared in the proof of Theorem 4.5, again. If $|\Gamma(s)| > 1$, then $|\Gamma^+(s) \cup \Gamma^-(s)| > 1$ holds in D. By Theorem 4.4, a strongly splittable pair $\{e' = us, f' = sv\}$ in D can be chosen so that $u \neq v$. This pair corresponds to a strongly splittable pair $\{e = us, f = sv\}$ in G with $u \neq v$, as required. □

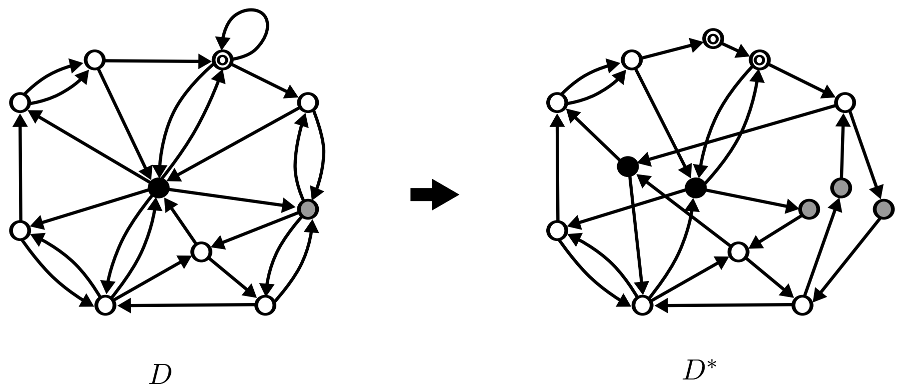


Figure 4.2: An admissible detachment D^* of a digraph D

4.3 Eulerian detachments of digraphs

In this section, we consider Eulerian digraphs D which may have loops. We call a degree specification $g = (\mathcal{V}, \rho^+, \rho^-)$ for D *even* if $\rho^+(x) = \rho^-(x)$ for all $x \in V^*$, and we may denote ρ^+ and ρ^- by ρ in this case. In the following, we show that there exists a g-detachment of D that satisfies a local edge-connectivity requirement for any even degree specification g.

For a digraph $D = (V, A)$ and a degree specification g (possibly not even), let

$$r_g(x, y) = \min\{\rho^+(x), \rho^-(y), \lambda(u, v; D)\}$$

if $x \in V_u$ and $y \in V_v$ for some $u, v \in V$, where we define $\lambda(u, v; D) = +\infty$ if $u = v$. Note that it holds $\lambda(x, y; D^*) \leq r_g(x, y)$ for any g-detachments D^* and $x, y \in V^*$. We call a g-detachment D^* of D *admissible* if D^* is r_g-edge-connected, i.e., $\lambda(x, y; D^*) \geq r_g(x, y)$ for all $x, y \in V^*$. This means that admissible g-detachments preserve the local edge-connectivity as much as possible. Figure 4.2 shows an example of admissible detachments of a digraph. The admissibility is defined also for $g(s)$-detachments since $g(s)$-detachments form a subclass of g-detachments. By proving the existence of admissible g-detachments for even degree specification g, we show a necessary and sufficient condition for a digraph to have an r-edge-connected g-detachment.

Lemma 4.3. *Let $D = (V, A)$ be an Eulerian digraph, and g be an even degree specification consisting of $\{V_v \mid v \in V\}$ and $\rho : V^* = \cup_{v \in V} V_v \to \mathbb{Z}_+$. Then there exists an admissible g-detachment of D.*

Proof. In the following, we show how to construct an admissible g-detachment for an arbitrary g. For this, it suffices to consider constructing an admissible $g(s)$-detachment for $s \in V$ since splitting all vertices $v \in V$ into V_v preserving admissibility finally gives an admissible g-detachment of G.

Suppose that $V_s = \{s_1, \ldots, s_n\}$ and that we have already obtained an admissible detachment $D_i = \{V \cup \{s_1, \ldots, s_i\}, A_i\}$ of D such that

$$\deg(x; D_i) = \begin{cases} \deg(x; D) & \text{if } x \in V - s, \\ \rho(s_j) & \text{if } x = s_j \text{ with } 1 \leq j \leq i, \\ \sum_{j=i+1}^{n} \rho(s_j) & \text{if } x = s, \end{cases}$$

where we denote $d^+(x; D_i) = d^-(x; D_i)$ by $d(x; D_i)$ because D_i is Eulerian. Note that

$$\lambda(x, y; D_i) = \begin{cases} r_g(x, y) & \text{if } \{x, y\} \subseteq V \cup \{s_1, \ldots, s_i\} - s, \\ \min\{\deg(s; D_i), \lambda(x, y; D)\} & \text{if } s \in \{x, y\} \text{ and } \{x, y\} \subset V, \\ \min\{\deg(s; D_i), \rho(s_j)\} & \text{if } \{x, y\} = \{s, s_j\} \text{ with } 1 \le j \le i \end{cases}$$

holds by admissibility. Below, we show how to construct an admissible detachment $D_{i+1} = \{V \cup \{s_1, \ldots, s_{i+1}\}, A_{i+1}\}$ from D_i such that $\deg(s_j; D_{i+1}) = \rho(s_j)$ for $j = 1, \ldots, i+1$ and $\deg(s; D_{i+1}) = \sum_{j=i+2}^{n} \rho(s_j)$. This inductively proves the lemma since $D_n - s$ is an admissible $g(s)$-detachment of D (notice that $\deg(s; D_n) = 0$).

First, prepare $D' = (V \cup \{s_1, \ldots, s_{i+1}\}, A_i \cup A')$ from D_i by adding a new vertex s_{i+1} and an arc set A' consisting of $\rho(s_{i+1})$ arcs ss_{i+1} and $\rho(s_{i+1})$ arcs $s_{i+1}s$. Then it holds

$$\deg(x; D') = \begin{cases} \deg(x; D_i) = \deg(x, D) & \text{if } x \in V - s, \\ \deg(s_j; D_i) = \rho(s_j) & \text{if } x = s_j \text{ with } 1 \le j \le i, \\ \rho(s_{i+1}) & \text{if } x = s_{i+1}, \\ \deg(s; D_i) + \rho(s_{i+1}) = 2\rho(s_{i+1}) + \sum_{j=i+2}^{n} \rho(s_j) & \text{if } x = s. \end{cases}$$

Moreover, $\lambda(x, y; D') = \lambda(x, y; D_i)$ is obvious if $s_{i+1} \notin \{x, y\}$. If $s_{i+1} \in \{x, y\}$, it holds $\lambda(x, y; D') = \min\{\rho(s_{i+1}), \lambda(s, z; D_i)\}$, where $z = \{x, y\} - s_{i+1}$ and $\lambda(s, s; D_i) = +\infty$. Hence for such $\{x, y\}$ (i.e., $s_{i+1} \in \{x, y\}$),

$$\lambda(x, y; D') = \begin{cases} \min\{\rho(s_{i+1}), \lambda(s, z; D)\} = r_g(s_{i+1}, z) & \text{if } \{x, y\} - s_{i+1} = z \in V - s, \\ \min\{\rho(s_{i+1}), \rho(s_j)\} = r_g(s_j, s_{i+1}) & \text{if } \{x, y\} = \{s_j, s_{i+1}\} \text{ with } 1 \le j \le i, \\ \min\{\rho(s_{i+1}), +\infty\} = \rho(s_{i+1}) & \text{if } \{x, y\} = \{s, s_{i+1}\} \end{cases}$$

holds, where we used $\rho(s_{i+1}) \le \deg(s; D_i)$ here. For each new arc ss_{i+1}, there is an arc zs such that $\{ss_{i+1}, zs\}$ is strongly splittable and $z \ne s_{i+1}$ by Theorems 4.3 and 4.4, while z is possibly s if exists. Splitting such a pair decreases the in- and out-degree of s by 1 respectively while preserving the local edge-connectivity between any pair of vertices in $V \cup \{s_1, \ldots, s_{i+1}\} - s$, and between s and the other vertices up to degree of s after splitting. Analogously for each new arc $s_{i+1}s$, there is an arc sz such that $\{s_{i+1}s, sz\}$ is strongly splittable and $z \ne s_{i+1}$. Let D_{i+1} be the graph obtained by splitting such pairs successively. Then D_{i+1} is a detachment of D. Moreover it holds

$$\deg(x; D_{i+1}) = \begin{cases} \deg(x; D') = \deg(x, D) & \text{if } x \in V - s, \\ \deg(s_j; D') = \rho(s_j) & \text{if } x = s_j \text{ with } 1 \le j \le i+1, \\ \deg(s; D') - 2\rho(s_{i+1}) = \sum_{j=i+2}^{n} \rho(s_j) & \text{if } x = s. \end{cases}$$

Furthermore, it also hold $\lambda(x, y; D_{i+1}) = \lambda(x, y; D')$ if $s \notin \{x, y\}$, and $\lambda(x, y; D_{i+1}) = \min\{d(s; D_{i+1}), \lambda(x, y; D')\}$ otherwise. This means

$$\lambda(x, y; D_{i+1}) = \begin{cases} r_g(x, y) & \text{if } \{x, y\} \subseteq V \cup \{s_1, \ldots, s_{i+1}\} - s, \\ \min\{d(s; D_{i+1}), \lambda(x, y; D)\} & \text{if } s \in \{x, y\} \text{ and } \{x, y\} \subseteq V, \\ \min\{\deg(s; D_{i+1}), \rho(s_j)\} & \text{if } \{x, y\} = \{s, s_j\} \text{ with } 1 \le j \le i+1. \end{cases}$$

Hence D_{i+1} is admissible, as required. $\square$

If an original digraph has some loops, its detachments may have loops as well. For undirected graphs, H. Nagamochi [61] showed a sufficient condition for an undirected graph to have a loopless connected g-detachment. In addition to this, we can see that there exists loopless k-edge-connected g-detachments if k is even and g satisfies a simple necessary condition by considering the proof of the theorem by C. St. J. A. Nash-Williams [62] (although we will not state the detail here). We extend our result in the above to loopless Eulerian g-detachments preserving local edge-connectivity both for digraphs, from which undirected graph version follows in the next section.

Lemma 4.4. *Let $D = (V, A)$ be an Eulerian digraph, and g be an even degree specification consisting of $\{V_v \mid v \in V\}$ and $\rho : V^* = \cup_{v \in V} V_v \to \mathbb{Z}_+$. Then D has a loopless admissible g-detachment if and only if $2\rho(x) \le d(v, v; D) + 2d(v, V - v; D)$ for all $v \in V$ and $x \in V_v$.*

Proof. First, we show necessity. Let us suppose that there exists a loopless admissible g-detachment D^* of D. Consider a new vertex $x \in V_v$ for a vertex $v \in V$. Trivially it hold $d(x, V^* - V_v; D^*) \le d(v, V - v; D)$ and $d(V^* - V_v, x; D^*) \le d(V - v, v; D)$. Since every arc between x and $V_v - x$ in D^* is originally a loop in D incident to v, it holds $d(x, V_v; D^*) + d(V_v, x; D^*) \le d(v, v; D)$. By $d^+(x; D^*) = d(x, V_v; D^*) + d(x, V^* - V_v; D^*)$ and $d^-(x; D^*) = d(V_v, x; D^*) + d(V^* - V_v, x; D^*)$, it holds that

$$\begin{aligned} 2\rho(x) &= d^+(x; D^*) + d^-(x; D^*) \\ &= d(x, V_v; D^*) + d(x, V^* - V_v; D^*) + d(V_v, x; D^*) + d(V^* - V_v, x; D^*) \\ &\le d(v, v; D) + 2d(v, V - v; D), \end{aligned}$$

implying the necessity.

In the next, we show sufficiency. We consider constructing an admissible $g(s)$-detachment of D. We have already shown that this can be done by an operation described in the proof of Lemma 4.3. Let us consider this again. If some loops are incident to s in $D' = (V \cup \{s_1, \ldots, s_i, s_{i+1}\}, A_i \cup A')$, pairs $\{ss, ss_{i+1}\}$ and $\{ss, s_{i+1}s\}$ are strongly splittable because splitting such a pair is equivalent to deleting one loop incident to s. At splitting on s in order to obtain D_{i+1}, we first continue choosing one of such pairs as long as some loops are incident to s. Then, no loops incident to s remain in D_{n-1} (and hence in D_n) by the following reason; It holds $\sum_{i=1}^{n} \rho(s_i) = \deg(s; D) = d(s, s; D) + d(s, V - s; D)$ by the hypothesis. Since $2\rho(s_n) \le d(s, s; D) + 2d(s, V - s; D)$, it holds that $\sum_{i=1}^{n-1} 2\rho(s_i) \ge d(s, s; D)$, which implies the above claim. If no loops are incident to s, we choose other strongly splittable pairs $\{xs, ss_{i+1}\}$ or $\{sx, s_{i+1}s\}$ such that $x \ne s_{i+1}$. This operation generates no loop obviously. Hence we can construct an admissible $g(s)$-detachment such that no loop is incident to a vertex in V_s, and therefore a loopless g-detachment. □

Theorem 4.7. *Let $D = (V, A)$ be an Eulerian digraph, and g be an even degree specification consisting of $\{V_v \mid v \in V\}$ and $\rho : V^* = \cup_{v \in V} V_v \to \mathbb{Z}_+$. Then there exists an r-edge-connected g-detachment of D if and only if $\lambda(u, v; D) \ge r(x, y)$ for all $x \in V_u$ and $y \in V_v$ with $u \ne v$ and $\rho(x) \ge r(x, y)$ for all $x \in V^*$ and $y \in V^* - x$. Such a g-detachment can be constructed without generating any loop if and only if $2\rho(x) \le d(v, v; D) + 2d(v, V - v; D)$ for all $v \in V$ and $x \in V_v$.*

Proof. First, let us consider the former part. Necessity is obvious. We can also derive the sufficiency from Lemma 4.3, since admissible detachments are r-edge-connected; i.e., $r_g(x,y) \geq r(x,y)$ for all $x, y \in V^* = \cup_{v \in V} V_v$, if $\lambda(u,v;D) \geq r(x,y)$ for $x \in V_u$ and $y \in V_v$ with $u \neq v$ and $\rho(x) \geq r(x,y)$ for $x \in V^*$ and $y \in V^* - x$.

Next, we consider the latter part. Necessity is proven as in the same way with Lemma 4.4. Sufficiency is derived from the existence of loopless admissible detachments, which is proven in Lemma 4.4. □

4.4 Eulerian detachments of undirected graphs

In this section, we consider Eulerian undirected graphs G which may have loops, and show the existence of g-detachments of G for any even degree specification $g = (\mathcal{V}, \rho)$, where g is called *even* if $\rho(x)$ is even for all $x \in V^*$. For an undirected graph, *admissibility* of g-detachments is defined in a similar way to digraphs, where r_g is defined as

$$r_g(x,y) = \min\{\rho(x), \rho(y), \lambda(x,y;G)\},$$

where we let $\lambda(x,y;G) = +\infty$ if $x = y$. We can derive the existence of admissible detachments for undirected graphs from that for digraphs.

Lemma 4.5. *Let $G = (V,E)$ be an Eulerian undirected graph, and g be an even degree specification consisting of $\{V_v \mid v \in V\}$ and $\rho : V^* = \cup_{v \in V} V_v \to \mathbb{Z}_+^{\mathrm{ev}}$. Then there exists an admissible g-detachment of G.*

Proof. Let $D = (V,A)$ be an orientation of G such that $2\lambda(u,v;D) = \lambda(u,v;G)$ for all $u, v \in V$. Moreover let g' be an even degree specification for D consisting of $\{V_v \mid v \in V\}$ and $\rho' : V^* \to \mathbb{Z}_+$ with $2\rho'(x) = \rho(x)$ for all $x \in V^*$. Notice that $2r_{g'}(x,y) = r_g(x,y)$ holds for all $x, y \in V^*$ by the definition of ρ' and by $2\lambda(u,v;D) = \lambda(u,v;G)$ for all $u, v \in V$.

By Lemma 4.3, there exists an admissible g'-detachment D^* of D. That is to say, $\lambda(x,y;D^*) \geq r_{g'}(x,y)$ for all $x, y \in V^*$. Let G^* be the underlying undirected graph of D^*. Since D^* is Eulerian, it holds $\lambda(x,y;G^*) = 2\lambda(x,y;D^*)$ for all $x, y \in V^*$. Hence we have

$$\lambda(x,y;G^*) = 2\lambda(x,y;D^*) \geq 2r_{g'}(x,y) = r_g(x,y)$$

for all $x, y \in V^*$, implying that G^* is an admissible g-detachment of G, as required. □

Lemma 4.6. *Let $G = (V,E)$ be an Eulerian undirected graph, and g be an even degree specification consisting of $\{V_v \mid v \in V\}$ and $\rho : V^* = \cup_{v \in V} V_v \to \mathbb{Z}_+^{\mathrm{ev}}$. Then G has a loopless admissible g-detachment if and only if $\rho(x) \leq d(v,v;G) + d(v,V-v;G)$ for all $v \in V$ and $x \in V_v$.*

Proof. Define D, D^*, g' and G^* as in the proof of Lemma 4.5. It hold $d(v,v;G) = c(v,v;D)$ and $d(v,V–v;G) = 2c(v,V–v;D)$. Hence $d(v,v;G)+d(v,V–v;G) = d(v,v;D)+2d(v,V–v;D)$ holds. Moreover we defined $\rho(x) = 2\rho'(x)$ for all $x \in V^*$. Hence the condition of $\rho(x) \leq$

$d(v,v;G)+d(v,V-v;G)$ is equivalent to $2\rho'(x) \le d(v,v;D)+2d(v,V-v;D)$ for all $v \in V$ and $x \in V_v$. Since G^* is loopless if and only if D^* is loopless, the lemma holds by the above fact and Lemma 4.4. $\square$

Theorem 4.8. *Let $G=(V,E)$ be an Eulerian undirected graph, and g be an even degree specification consisting of $\{V_v \mid v \in V\}$ and $\rho : V^* = \cup_{v \in V} V_v \to \mathbb{Z}_+^{\mathrm{ev}}$. Then there exists an r-edge-connected g-detachment of G if and only if $\lambda(u,v;G) \ge r(x,y)$ for all $x \in V_u$ and $y \in V_v$ with $u \ne v$ and $\rho(x) \ge r(x,y)$ for all $x \in V^*$ and $y \in V^* - x$. Such a g-detachment can be constructed without generating any loop if and only if $\rho(x) \le d(v,v;G)+d(v,V-v;G)$ for all $v \in V$ and $x \in V_v$.*

Proof. First, let us consider the former part. Necessity is obvious. We can also derive the sufficiency from Lemma 4.5, since admissible detachments are r-edge-connected; i.e., $r_g(x,y) \ge r(x,y)$ for all $x,y \in V^* = \cup_{v\in V} V_v$, if $\lambda(u,v;G) \ge r(x,y)$ for $x \in V_u$ and $y \in V_v$ with $u \ne v$ and $\rho(x) \ge r(x,y)$ for $x \in V^*$ and $y \in V^* - x$.

Next, we consider the latter part. Necessity is proven as in the same way with Lemma 4.4. Sufficiency is derived from the existence of loopless admissible detachments, which is proven in Lemma 4.6. $\square$

4.5 $\{3, d(s)-3\}$-detachment

In this section, we consider a degree specification $g(s) = (V_s, \rho)$ for an undirected graph such that $V_s = \{s, s'\}$, $\rho(s) = \deg(s) - 3$ and $\rho(s') = 3$. We assume that no cut-edge is incident to s and $\deg(s) \ge 6$. Then it holds $\lambda(u,v) \ge 2$ for any $u, v \in \Gamma(s)$ since otherwise cut-edge is incident to s. This implies that s and its all neighbors belong to the same 2-edge-connected component. Hence we can assume without loss of generality that a given graph is 2-edge-connected.

We define a function $r_g : \binom{V \cup s'}{2} \to \mathbb{Z}_+$ as follows;

$$r_g(x,y) = \begin{cases} \lambda(x,y;G) & x,y \in V-s, \\ \min\{\lambda(x,y;G), d(s;G)-3\} & s \in \{x,y\}, \{x,y\}-s \subseteq V-s, \\ \min\{\lambda(x,y;G), 3\} & s' \in \{x,y\}, \{x,y\}-s' \subseteq V-s, \\ 3 & \{x,y\} = \{s,s'\}. \end{cases}$$

For $\emptyset \ne X \subseteq V - s$, let

$$f_{r_g}(X) = \max_{x \in X, y \in V \cup s' - X} r_g(x,y) \quad \text{and} \quad h(X) = d(X;G) - f_{r_g}(X).$$

Notice that $\max_{x\in X, y \in V\cup s'-X} r_g(x,y) = \max_{x \in X, y \in V-X} r_g(x,y)$ since $r_g(x,s) \ge r_g(x,s')$ for all $x \in V-s$. Moreover note that $h(X) \ge 0$ for every $\emptyset \ne X \subseteq V-s$. A set $X \subseteq V-s$ is called *tight* if $h(X)=0$, *dangerous* if $h(X) \le 1$, and *bad* if $h(X) \le 2$. Similarly for Lemma 4.2, function h has weak submodularity as follows.

Lemma 4.7. *For every pair $X, Y \subseteq V - s$, at least one of*

$$h(X) + h(Y) \geq h(X \cap Y) + h(X \cup Y) + 2d(X - Y, Y - X; G), \tag{4.8}$$

and

$$h(X) + h(Y) \geq h(X - Y) + h(Y - X) + 2d(X \cap Y, V - (X \cup Y); G). \tag{4.9}$$

holds.

Proof. In Theorem 2.4, we have already seen that

$$d(X; G) + d(Y; G) = d(X \cap Y; G) + d(X \cup Y; G) + 2d(X - Y, Y - X; G), \tag{2.13}$$

and

$$d(X; G) + d(Y; G) = d(X - Y; G) + d(Y - X; G) + 2d(X \cap Y, V - (X \cup Y); G) \tag{2.14}$$

hold. On the other hand, f_{r_g} is weakly supermodular by Theorem 2.5, i.e., at least one of

$$f_{r_g}(X) + f_{r_g}(Y) \leq f_{r_g}(X \cap Y) + f_{r_g}(X \cup Y) \tag{4.12}$$

and

$$f_{r_g}(X) + f_{r_g}(Y) \leq f_{r_g}(X - Y) + f_{r_g}(Y - X) \tag{4.13}$$

holds. If (4.12) holds, we obtain (4.8) by subtracting (4.12) from (2.13). If (4.13), we obtain (4.9) by subtracting (4.13) from (2.14), as required. □

Lemma 4.8. *No bad set contains all neighbors of s.*

Proof. Let us consider a bad set X with $\Gamma(s) \subseteq X$. If $f_{r_g}(X) = r_g(u, s)$ for some $u \in X$, then $d(X; G) \geq d(s; G) \geq r_g(u, s) + 3 = f_{r_g}(X) + 3$, i.e., $h(X) \geq 3$, a contradiction to the badness of X. Otherwise, $f_{r_g}(X) = r_g(u, v)$ for some $u \in X$ and $v \in V - X$. Then $d(X; G) = d(X, V - X; G) + d(s) \geq r_g(u, v) + 6$, i.e., $h(X) \geq 6$, a contradiction again. □

Lemma 4.9. *For a tight set X, $d(X, s; G) \leq d(s; G) - 2$.*

Proof. Lemma 4.8 implies that $d(X, s; G) < d(s; G)$. So let us suppose $d(X, s; G) = d(s; G) - 1$. If $f_{r_g}(X) = r_g(u, s)$ for some $u \in X$, then $d(X; G) \geq d(s; G) - 1 \geq r_g(u, s) + 2 = f_{r_g}(X) + 2$, a contradiction to the tightness of X. Otherwise, $f_{r_g}(X) = r_g(u, v)$ for some $u \in X$ and $v \in V - X$. Then $f_{r_g}(X) = r_g(u, v) = \lambda(u, v) \leq 1 + d(X, V - (X \cup s); G) = 1 + d(X; G) - d(X, s; G) = d(X; G) - d(s; G) + 2 \leq d(X; G) - 4$, i.e., $h(X) \geq 4$, a contradiction again. □

For a set $\{su, sv, sz\}$ of edges incident to s, $G^* = (V \cup s', E \cup \{s'u, s'v, s'z\} - \{su, sv, sz\})$ is a $g(s)$-detachment of G. We call such a set of edges *admissible* if G^* is r_g-edge-connected. We can characterize admissible sets as follows.

Lemma 4.10. *Set $\{e_1, e_2, e_3\}$ of edges incident to s is admissible if and only if all of the following conditions hold;*

(a) *There is no tight set $X \subseteq V - s$ with $|\{e_1, e_2, e_3\} \cap \delta(X)| \geq 2$.*

(a') *There is no set* $X \subseteq V$ *with* $|\{e_1, e_2, e_3\} \cap \delta(X)| \geq 2$, $h(X) = 1$ *and* $f_{r_g}(X) = 2$.

(b) *There is no bad set* $X \subseteq V - s$ *with* $|\{e_1, e_2, e_3\} \cap \delta(X)| = 3$.

(b') *There is no set* $X \subseteq V$ *with* $|\{e_1, e_2, e_3\} \cap \delta(X)| = 3$, $h(X) = 3$ *and* $f_{r_g}(X) = 2$.

Proof. First, let us prove the necessity. Suppose (a) does not hold, that is, there exists a tight set X with with $|\{e_1, e_2, e_3\} \cap \delta(X)| \geq 2$. For any set $Y \subseteq V - s$, it holds $f_{r_g}(Y) \leq f_{r_g}(Y + s')$ since $r_g(v, s') \leq r_g(v, s)$. Hence

$$d(X + s'; G^*) \leq d(X; G^*) - 1 = f_{r_g}(X) - 1 \leq f_{r_g}(X + s') - 1,$$

implying that G^* is not r_g-edge-connected.

Suppose (a') does not hold, that is, there exists a set X with with $|\{e_1, e_2, e_3\} \cap \delta(X)| \geq 2$, $h(X) = 1$, and $f_{r_g}(X) = 2$. This means that $d(X; G^*) = d(X; G) = 3$ and $d(X + s'; G^*) \leq 2$. Moreover, $f_{r_g}(X + s') = 3$ because $r_g(s, s') = 3$. Hence

$$d(X + s'; G^*) < f_{r_g}(X + s'),$$

implying that G^* is not r_g-edge-connected.

Suppose (b) does not hold, that is, there exists a bad set X with with $|\{e_1, e_2, e_3\} \cap \delta(X)| = 3$. Then

$$d(X + s'; G^*) = d(X; G^*) - 3 = d(X; G) - 3 \leq f_{r_g}(X) - 1 \leq f_{r_g}(X + s') - 1,$$

implying that G^* is not r_g-edge-connected.

Suppose (b') does not hold, that is, there exists a set X with with $|\{e_1, e_2, e_3\} \cap \delta(X)| = 3$, $h(X) = 3$, and $f_{r_g}(X) = 2$. This means that $d(X; G) = d(X; G^*) = 5$ and $d(X + s'; G^*) = 2$. Since $f_{r_g}(X + s') = 3$, it holds

$$d(X + s'; G^*) < f_{r_g}(X + s'),$$

implying that G^* is not r_g-edge-connected.

In the next, we show the sufficiency. Suppose $\{e_1, e_2, e_3\}$ is not admissible. Then there exists a set $Y \subset V + s'$ with $d(Y; G^*) < f_{r_g}(Y)$. We can assume without loss of generality that $s \notin Y$. Then, $s' \in Y$ since otherwise we would have a contradiction that $h(Y) = d(Y; G) - f_{r_g}(Y) = d(Y; G^*) - f_{r_g}(Y) < 0$.

We can show that $|\{e_1, e_2, e_3\} \cap \delta(Y)| \neq 1$ since otherwise we would have

$$d(Y - s'; G) = d(Y - s'; G^*) = d(Y; G^*) - 1 < f_{r_g}(Y) - 1 \leq f_{r_g}(Y - s').$$

Second inequality (i.e., $f_{r_g}(Y) \leq f_{r_g}(Y - s') + 1$) can be derived from the facts that $r_g(s', v) \leq 3$ for every $v \in V$ and that $r_g(u, v) \geq 2$ for every pair $u, v \in V + s'$. In a similar way, we can also see that $\{e_1, e_2, e_3\} \cap \delta(Y) \neq \emptyset$ since otherwise we would have

$$d(Y - s'; G) = d(Y - s'; G^*) = d(Y; G^*) - 3 < f_{r_g}(Y) - 3 \leq f_{r_g}(Y - s') - 2.$$

Hence $|\{e_1, e_2, e_3\} \cap \delta(Y)|$ is either 2 or 3.

Suppose $|\{e_1, e_2, e_3\} \cap \delta(Y)| = 2$. Then,

$$d(Y - s'; G) = d(Y - s'; G^*) = d(Y; G^*) + 1 \leq f_{r_g}(Y).$$

If $f_{r_g}(Y) = f_{r_g}(Y - s')$, then $d(Y - s'; G) \leq f_{r_g}(Y - s')$, implying that (a) does not hold. Otherwise (i.e., $f_{r_g}(Y) = f_{r_g}(Y - s') + 1$), it holds $d(Y - s'; G) \leq f_{r_g}(Y - s') + 1$. When equality of this inequality does not hold, (a) does not hold. Hence let $d(Y - s'; G) = f_{r_g}(Y - s') + 1$. Notice that it must hold $f_{r_g}(Y - s') = 2$ if $f_{r_g}(Y) = f_{r_g}(Y - s') + 1$. Hence (a') does not hold.

Suppose $|\{e_1, e_2, e_3\} \cap \delta(Y)| = 3$. Then

$$d(Y - s'; G) = d(Y - s'; G^*) = d(Y; G^*) + 3 < f_{r_g}(Y) + 3.$$

If $f_{r_g}(Y) = f_{r_g}(Y - s')$, then $d(Y - s'; G) \leq f_{r_g}(Y - s') + 2$, implying that (b) does not hold. Otherwise (i.e., $f_{r_g}(Y) = f_{r_g}(Y - s') + 1$), it holds $d(Y - s'; G) \leq f_{r_g}(Y - s') + 3$. When equality of this inequality does not hold, (b) does not hold. Hence let $d(Y - s'; G) = f_{r_g}(Y - s') + 3$. Note that $f_{r_g}(Y - s') = 2$. Then (b') does not hold. $\square$

The following two lemmas show that conditions (a') and (b') in Lemma 4.10 can be removed.

Lemma 4.11. *If* (a) *holds, then* (a') *holds.*

Proof. Suppose that (a') does not hold, i.e., there exists a set $X \subseteq V - s$ with $|\{e_1, e_2, e_3\} \cap \delta(X)| \geq 2$, $d(X; G) = 3$ and $f_{r_g}(X) = 2$. In the following, we show that both (a) does not hold, i.e., there exists a set $Y \subseteq V - s$ with $|\{e_1, e_2, e_3\} \cap \delta(Y) \geq 2$ and $h(Y) = 0$. For this, let us conversely assume that no such Y exists.

Since $f_{r_g}(X) = 2$, $\lambda(s, v; G) = 2$ holds for every $v \in X$. Hence there exists a set $Z_v \subseteq V - s$ with $d(Z_v; G) = 2$ and $v \in Z_v$ for every $v \in X$. If $d(X \cup Z_v) = 2$, then $X \cup Z_v$ gives the above Y. Hence let $d(X \cup Z_v) \geq 3$. Then we can assume that $Z_v \subseteq X$ since otherwise $d(X \cap Z_v) = 2$ would hold by

$$3 + 2 = d(X; G) + d(Z_v; G) \geq d(X \cup Z_v; G) + d(X \cap Z_v; G) \geq 3 + 2,$$

implying that Z_v can be replaced by $X \cap Z_v$. Moreover, we can assume that either $Z_u \cap Z_v = \emptyset$ or $Z_u = Z_v$ holds for every $u, v \in X$ since otherwise

$$2 + 2 \geq d(Z_u; G) + d(Z_v; G) \geq d(Z_u \cup Z_v; G) + d(Z_u \cap Z_v; G) \geq 2 + 2,$$

holds, which would imply that Z_u and Z_v can be replaced by $Z_u \cup Z_v$. To sum up, we have a partition $Z_1, \ldots, Z_k$ of X with $d(Z_i; G) = 2$, $i = 1, \ldots, k$. Then, we have

$$d(X; G) = \sum_{i=1}^{k} d(Z_i; G) - 2 \sum_{1 \leq i < j \leq k} d(Z_i, Z_j; G).$$

Since the right hand side of this equality is even, $d(X; G)$ is even. This is a contradiction. $\square$

Lemma 4.12. *If* (b) *holds,* (b') *holds.*

Proof. Suppose that (b') does not hold; i.e., there exists a set $X \subseteq V-s$ with $|\{e_1,e_2,e_3\} \cap \delta(X)| = 3$, $d(X;G) = 5$ and $f_{r_g}(X) = 2$. In the following, we show that (b) does not hold, i.e., there exists a set $Y \subset V-s$ with $|\{e_1,e_2,e_3\} \cap \delta(X)| = 3$ and $h(Y) \leq 2$. For this, let us assume conversely that no such Y exists.

Since $f_{r_g}(X) = 2$, $\lambda(s,v;G) = 2$ holds for every $v \in X$. Hence there exists $Z_v \subseteq V-s$ with $d(Z_v;G) = 2$ and $v \in Z_v$ for every $v \in X$. If $d(X \cup Z_v) \leq 4$, then $X \cup Z_v$ gives the above Y. Hence $d(X \cup Z_v) \geq 5$ holds. Then we can assume that $Z_v \subseteq X$, since otherwise $d(X \cap Z_v) = 2$ would hold by

$$5+2 = d(X;G) + d(Z_v;G) \geq d(X \cup Z_v;G) + d(X \cap Z_v;G) \geq 5+2,$$

implying that Z_v can be replaced by $X \cap Z_v$. Moreover, we can assume that either $Z_u \cap Z_v = \emptyset$ or $Z_u = Z_v$ holds for every $u,v \in X$ since otherwise it would hold

$$2+2 \geq d(Z_u;G) + d(Z_v;G) \geq d(Z_u \cup Z_v;G) + d(Z_u \cap Z_v;G) \geq 2+2,$$

which implies that Z_u and Z_v can be replaced by $Z_u \cup Z_v$. To sum up, we have a partition $Z_1,\ldots,Z_k$ of X with $d(Z_i) = 2$. Then, we have

$$d(X;G) = \sum_{i=1}^{k} d(Z_i;G) - 2 \sum_{1 \leq i < j \leq k} d(Z_i,Z_j;G).$$

Since the right hand side of this equality is even, $d(X;G)$ is even. This is a contradiction. □

Lemma 4.10 can be simplified as follows.

Lemma 4.13. *Set $\{e_1,e_2,e_3\}$ of edges incident to s is admissible if and only if both of the following two conditions hold;*

(a) *There is no tight set $X \subseteq V-s$ with $|\{e_1,e_2,e_3\} \cap \delta(X)| \geq 2$.*

(b) *There is no bad set $X \subseteq V-s$ with $|\{e_1,e_2,e_3\} \cap \delta(X)| = 3$.*

Proof. Immediate from Lemmas 4.10, 4.11 and 4.12. □

The following property is useful for analysis.

Lemma 4.14. *Let $T \subseteq V$ be a tight set in $G=(V,E)$, and $\hat{G} = (\hat{V},\hat{E})$ be the graph obtained from G by contracting T into a single vertex t. If set $\{\hat{e}_1,\hat{e}_2,\hat{e}_3\}$ of edges in $\hat{E}$ incident to s is admissible, then set $\{e_1,e_2,e_3\}$ of edges in E corresponding to $\{\hat{e}_1,\hat{e}_2,\hat{e}_3\}$ is admissible.*

Proof. Since $\{e_1,e_2,e_3\}$ is not admissible in G, there exists a tight set $X \subseteq V-s$ with $|\{e_1,e_2,e_3\} \cap \delta(X)| \geq 2$ or a bad set $Y \subseteq V-s$ with $|\{e_1,e_2,e_3\} \cap \delta(Y)| = 3$ by Lemma 4.13. In the following, we show that $\{\hat{e}_1,\hat{e}_2,\hat{e}_3\}$ is not admissible in $\hat{G}$ in both cases.

First, we consider the former case, i.e.,there exists a tight set $X \subseteq V-s$ with $|\{e_1,e_2,e_3\} \cap \delta(X)| \geq 2$. By Lemma 4.7, it holds at least one of

$$0+0 \geq h(X) + h(T) \geq h(X \cap T) + h(X \cup T) + 2d(X-T,T-X) \tag{4.14}$$

and

$$0+0 \geq h(X)+h(T) \geq h(X-T)+h(T-X)+2d(X\cap T, V-(X\cup T)). \tag{4.15}$$

If (4.14) holds, we have $h(X \cup T) = 0$, which implies that $X \cup T$ is another tight set with $|\{e_1, e_2, e_3\} \cap \delta(X \cup T)| \geq 2$. Then $\{\hat{e}_1, \hat{e}_2, \hat{e}_3\}$ is not admissible because $X \cup t$ is a tight set in $\hat{G}$ with $|\{\hat{e}_1, \hat{e}_2, \hat{e}_3\} \cap \delta(X \cup t)| \geq 2$. If (4.15) holds, we have $h(X - T) = 0$. Moreover, $|\{e_1, e_2, e_3\} \cap (X - T)| \geq 2$ since $d(X \cap T, s) \leq d(X \cap T, V - (X \cup T)) = 0$. We hence can see that $\{\hat{e}_1, \hat{e}_2, \hat{e}_3\}$ is not admissible again because $X - T$ is a tight set in $\hat{G}$ with $|\{\hat{e}_1, \hat{e}_2, \hat{e}_3\} \cap \delta(X - T)| \geq 2$.

Next, we consider the latter case, i.e., there exists a bad set $Y \subseteq V - s$ with $|\{e_1, e_2, e_3\} \cap Y| = 3$. By Lemma 4.7, it holds at least one of

$$2+0 \geq h(Y)+h(T) \geq h(Y\cap T)+h(Y\cup T)+2d(Y-T, T-Y) \tag{4.16}$$

and

$$2+0 \geq h(Y)+h(T) \geq h(Y-T)+h(T-Y)+2d(Y\cap T, V-(Y\cup T)). \tag{4.17}$$

If (4.16) holds, we have $2 \geq h(Y \cup T)$, which implies that $Y \cup T$ is another bad set with $|\{e_1, e_2, e_3\} \cap \delta(Y \cup T)| = 3$. Then $\{\hat{e}_1, \hat{e}_2, \hat{e}_3\}$ is not admissible because $Y + t$ is also a bad set in $\hat{G}$ with $|\{\hat{e}_1, \hat{e}_2, \hat{e}_3\} \cap \delta(Y \cup t)| = 3$. If (4.17) holds, we have $2 \geq h(Y - T)$. Moreover, $|\{e_1, e_2, e_3\} \cap \delta(Y - T)| = 3$ since $d(Y \cap T, s) \leq d(Y \cap T, V - (Y \cup T)) = 0$. We hence can see that $\{\hat{e}_1, \hat{e}_2, \hat{e}_3\}$ is not admissible again because $Y - T$ is a bad set in $\hat{G}$ with $|\{\hat{e}_1, \hat{e}_2, \hat{e}_3\} \cap \delta(Y - T)| = 3$. □

By the above lemmas, we can derive the following result for the existence of g-detachments that preserves local edge-connectivity as much as possible.

Theorem 4.9. *Let s be a vertex in an undirected graph G such that no cut-edge is incident to s and $d(s; G) \geq 6$. Moreover let $g(s) = \{V_s, \rho\}$ be a degree specification such that $V_s = \{s, s'\}$, $\rho(s) = d(s; G) - 3$ and $\rho(s') = 3$. Then there exists an admissible $g(s)$-detachment of G if and only if $\Gamma(s)$ is not covered by exactly two tight sets.*

Proof. Let us suppose that $\Gamma(s)$ is covered by exactly two tight sets. Then any sets of three edges incident to s contain at least two edges entering one tight set, implying that there exists no admissible set by Lemma 4.13.

In what follows, we suppose that $\Gamma(s)$ is not covered by exactly two tight sets in order to show the necessity. Furthermore, we assume that every tight set is a singleton without loss of generality by Lemma 4.14. If $|\Gamma(s)| \leq 3$, a set of edges covering all neighbors of s is admissible by Lemmas 4.8, 4.9 and 4.13.

Let $|\Gamma(s)| \geq 5$. We let $u, v \in \Gamma(s)$ be a pair of neighbors of s such that $\{su, sv\}$ is a strongly splittable pair, whose existence is guaranteed by Theorem 4.2. By Theorem 4.1, no dangerous set contains both of u and v because $f_{r_G}(X) \geq f_{r_g}(X)$ for $X \subseteq V - s$. If there exists an admissible set containing u and v, we are done. Otherwise, there is a maximal bad set X_w with $u, v, w \in X_w$ for every $w \in \Gamma(s) - \{u, v\}$. For any distinct X_w and X_z with $w, z \in \Gamma(s) - \{u, v\}$, it does not hold (4.8) since

$$2+2 \geq h(X_w)+h(X_z) \geq h(X_w \cap X_z)+h(X_w \cup X_z) \geq 2+3,$$

where $h(X_w \cap X_z) \geq 2$ follows from $\{u, v\} \subseteq X_w \cap X_z$ and $h(X_w \cup X_z) \geq 3$ follows from the maximality of X_w and X_z. Hence it holds

$$2 + 2 \geq h(X_w) + h(X_z) \geq h(X_w - X_z) + h(X_z - X_w) \\ + 2d(X_w \cap Z_z, V - (X_w \cup Z_z)) \geq 0 + 0 + 4, \quad (4.18)$$

which implies that $X_w - X_z$ and $X_z - X_w$ are tight sets. By the assumption, it holds $|X_w - X_z| = |X_z - X_w| = 1$. Since $|\Gamma(s) - \{u, v\}| \geq 3$, there are at least three distinct maximal bad sets X_w, X_z and X_y containing u and v. Inequalities (4.18) for pairs of these sets imply that $d(X_w \cap X_z \cap X_y) = 2$. This implies $h(X_w \cap X_z \cap X_y) = 0$ since $f_{r_g}(X_w \cap X_z \cap X_y) \geq 2$ by the assumption that G is 2-edge-connected. This contradicts the definition of $\{u, v\}$.

Let $\Gamma(s) = \{u, v, w, z\}$, i.e., $|\Gamma(s)| = 4$. Suppose that no dangerous set contains both of $u, v \in \Gamma(s)$ again. If no admissible set exists, then there exists a maximal bad set X_x such that $\Gamma(s) - x \subseteq X_x$ and $x \notin X_x$ for any $x \in \Gamma(s)$ by Lemmas 4.8 and 4.13. It does not hold (4.8) for X_z and X_w since

$$2 + 2 \geq h(X_z) + h(X_w) \geq h(X_z \cap X_w) + h(X_z \cup X_w) \geq 2 + 3,$$

where $h(X_z \cap X_w) \geq 2$ follows from $\{u, v\} \subseteq X_z \cap X_w$ and $h(X_z \cup X_w) \geq 3$ follows from the maximality of X_z and X_w. Hence it holds

$$2 + 2 \geq h(X_z) + h(X_w) \geq h(X_z - X_w) + h(X_z - X_w) \\ + 2d(X_z \cap X_w, V - (X_z \cup X_w)) \geq 0 + 0 + 4,$$

which implies that $X_z - X_w = \{w\}$, $X_w - X_z = \{z\}$, $\delta(X_z \cap X_w, V - (X_z \cup X_w)) = \{su, sv\}$ and $d(s, u) = d(s, v) = 1$. First, let us suppose $f_{r_g}(X_u) = r_g(x, s)$ with $x \in X_u$. It holds $d((X_z \cap X_w) - X_u, X_u) \geq 1$ since otherwise $\delta((X_z \cap X_w) - X_u) = \{su\}$, implying that su is a cut-edge. Then, it holds $d(X_u) \geq d(s) - 1 + 1 \geq r_g(s, x) + 3 \geq f_{r_g}(X_u) + 3$, which contradicts $h(X_u) \leq 2$. Next, let us suppose $f_{r_g}(X_u) = r_g(x, y)$ for some $x \in X_u$ and $y \in V - s - X_u - (X_z \cap X_w)$. Then, it holds that

$$d(X_u) \geq d(s) + d(X_u, V - X_u - (X_z \cap X_w)) \geq 6 + \lambda(x, y) \geq f_{r_g}(X) + 6,$$

indicating a contradiction, again. Therefore, it must hold $f_{r_g}(X_u) = r(x, y)$ with $x \in X_u$ and $y \in (X_z \cap X_w) - X_u$. However, $\lambda(x, y) \leq d(X_u, X_z \cap X_w) + 1$ holds since $d((X_z \cap X_w) - X_u, V \cup s - (X_z \cap X_w) - X_u) = 1$. This implies that $h(X_u) \geq 4$ since $d(X_u) \geq d(X_u, X_z \cap X_w) + d(s) - 1 \geq \lambda(x, y) + 4 = f_{r_g}(X_u) + 4$, a contradiction. □

Chapter 5

Network Design with Edge-Connectivity and Degree Constraints

This chapter considers the problem of constructing a minimum cost multigraph with a specified edge-connectivity under degree constraints. In our algorithm to this problem, the results on splitting and detachment presented in Chapter 4 play an important role.

5.1 Introduction

It is a main concern in the field of network design to construct a graph of least cost which satisfies some connectivity requirement. Actually many results on this topic have been obtained so far. In this chapter, we consider a network design problem that asks to find a minimum cost r-edge-connected multigraph on a metric edge cost under degree constraints. Formally the problem we consider is formulated as follows.

> **Network design problem with edge connectivity and degree constraints**
> Given a vertex set V, a connectivity demand $r : \binom{V}{2} \to \mathbb{Z}_+$, a lower capacity $a : V \to \mathbb{Z}_+$, an upper capacity $b : V \to \mathbb{Z}_+$ and a metric edge cost $w : \binom{V}{2} \to \mathbb{Q}_+$, find a minimum cost multigraph $G = (V, E)$ with no loops such that $\lambda(u, v; G) \geq r(u, v)$ for each pair $u, v \in V$ and $a(v) \leq d(v; G) \leq b(v)$ for each $v \in V$.

We denote a problem instance consisting of the above inputs by (V, r, a, b, c).

This problem includes a wide range of classically fundamental problems. For example, let $a = 0$ and $b = +\infty$, i.e., no degree constraints are imposed. Then the problem is equivalent to the minimum spanning tree problem if $r = 1$ (i.e., $(V, 1, 0, +\infty, w)$). If $r(x, y) = 1$ for $x, y \in \mathcal{T} \subseteq V$ and $r(x, y) = 0$ otherwise, then the problem is the Steiner tree problem with some terminal set $\mathcal{T}$. This is an NP-hard problem and a ρ-approximation algorithm is given by G. Robins and A. Zelikovsky [67], where $\rho = 1 + (\ln 3)/2 \leq 1.55$. For a general r, $(V, r, 0, +\infty, w)$ is the Steiner network problem without edge-capacity, to which a

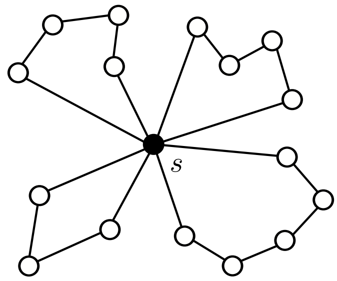


Figure 5.1: A solution for VRP with $m = 4$

2-approximation algorithm is proposed by K. Jain [45] as seen in Theorem 2.11. On the other hand, let $r = 0$, i.e., no connectivity demand is required. If $b = +\infty$, then the problem $(V, 0, a, +\infty, w)$ is exactly the a-edge cover problem (see Section 2.2.3). If $a = b$, i.e., the degree of each vertex is exactly specified, then the problem $(V, 0, b, b, w)$ is the perfect b-matching problem, which is especially called the perfect b-matching problem if $b = 1$. For general degree bounds a and b, the problem $(V, 0, a, b, w)$ is known to be solvable in a polynomial time (see [71] for example). As an example of instances that have both degree bounds and edge-connectivity demand, the *vehicle routing problem* (VRP) is known.

Vehicle routing problem (VRP)
Given an integer $m \in \mathbb{Z}_+$, a vertex set V including a designate vertex s (called *depot*) and a metric edge cost $w : \binom{V}{2} \to \mathbb{Q}_+$, find a minimum cost set of m cycles, each containing s, such that each of the other vertices is covered by exactly one cycle.

Figure 5.1 shows a feasible solution for an instance of the VRP with $m = 4$. Notice that the VRP is equivalent to $(V, 2, a, b, w)$, where $a(s) = b(s) = 2m$ and $a(v) = b(v) = 2$ for $v \in V - s$. The VRP contains the metric TSP as a special case with $m = 1$. As seen in Section 2.4, TSP with general edge cost cannot be approximated unless P = NP, and metric TSP is approximable within a factor of 1.5. For VRP, a primal-dual algorithm achieves 2-approximation [23, 32]. In addition to these, S. P. Fekete et al. [18] considered the problem $(V, 1, 0, b \geq 2, w)$, i.e., a problem to find a minimum cost spanning tree under the constraint that the degree of each vertex is bounded from the above. They proved that the problem $(V, 1, 0, b \geq 2, w)$ is approximable within a factor of $2 - \min_{v \in V, d(v;T) > 2} (b(v) - 2)/(d(v; T) - 2)$, where T is a minimum spanning tree. The Steiner tree problem with a further constraint such that all terminals must be leaves (i.e., $b(v) = 1$ for all $v \in \mathcal{T}$ and $b(v) = +\infty$ for all $v \in V - \mathcal{T}$) is called the terminal Steiner tree problem (or full Steiner tree problem). Notice that assuming the metric edge cost does not preserve generality in contrast to the Steiner tree problem. D. E. Drake and S. Hougardy [15] showed that this problem with metric edge cost can be approximated within 2ρ if the Steiner tree problem can be approximated within ρ.

Although the problem (V, r, a, b, w) is a natural framework as a generalization of the above problems, a few results on this problem setting have been obtained so far. A. Frank [21] solved the problem of augmenting a given graph to an r-edge-connected graph by adding a smallest number of new edges under lower and upper bounds on degrees. This implies that problem $(V, r, a, b, 1)$ admits a polynomial time algorithm. Moreover, an extended result by A. Frank [21] suggests that (V, r, a, b, w) is polynomially solvable in a special case where cost $w(e)$ for each edge $e = uv$ is given by $\nu(u) + \nu(v)$ for some vertex weight $\nu : V \to \mathbb{Q}_+$.

In the subsequent sections, we show several conditions on functions r, a, b and w for which the problem admits an approximation algorithm. To design most algorithms proposed in this paper, splitting and detachment introduced in Chapter 4 play an important role.

5.2 Problem with lower capacity

In this section, we consider the problem $(V, r, a, +\infty, w)$. As stated in Section 5.1, this problem is equivalent to the Steiner network problem if $a(v) = 0$ for all $v \in V$, and is shown to be 2-approximable by K. Jain (see Section 2.3.2). In what follows, we show that Jain's algorithm can be applied to the problem $(V, r, a, +\infty, w)$ even if $a(v) > 0$ for some $v \in V$. For this, observe that the problem finds a minimum cost loopless multigraph G such that $d(X; G) \geq f(X)$ for every $X \subseteq V$, where

$$f(X) = \begin{cases} 0 & \text{if } X = \emptyset \text{ or } V, \\ \max\{a(u), \max_{v \in V-u} r(u, v)\} & \text{if } X = \{u\} \text{ or } V - \{u\}, \\ \max_{u \in X, v \in V-X} r(u, v) & \text{otherwise.} \end{cases} \tag{5.1}$$

Now let us see that function f is weakly supermodular, which implies that $(V, r, a, +\infty, w)$ is 2-approximable.

Theorem 5.1. *$(V, r, a, +\infty, w)$ is approximable within a factor of 2.*

Proof. By Theorem 2.11 and the above observation, it suffices to show that set function f defined in (5.1) is weakly supermodular. Let $X \subseteq V$ and $Y \subseteq V$. It is an easy to see that, if $X \subseteq Y$ or $Y \subseteq X$, then (2.17) holds. Similarly (2.18) holds if $X \cap Y = \emptyset$. If $X \cup Y = V$, then (2.18) holds, since f is symmetric (i.e., $f(X) = f(V - X)$ for every $X \subseteq V$) and we have $f(X) + f(Y) = f(V - X) + f(V - Y) = f(Y - X) + f(X - Y)$. Then we only need to consider the case in which each of $X - Y$, $Y - X$, $X \cap Y$ and $V - (X \cup Y)$ is non-empty. In this case, $|X| \notin \{1, |V| - 1\}$ and $|Y| \notin \{1, |V| - 1\}$ hold, and inequality (2.17) or (2.18) follows from the weakly supermodularity of $f_r(X) = \max_{u \in X, v \in V-X} r(u, v)$. $\square$

5.3 Problem with upper capacity

In this section, we discuss the approximability of the problem $(V, r \geq 2, 0, b, w)$. Notice that the problem has no feasible solution if there is a vertex $v \in V$ with $b(v) < \max_{u \in V-v} r(u, v)$. Therefore we suppose that $b(v) \geq \max_{u \in V-v} r(u, v)$ for each $v \in V$. In addition, we can assume without loss of generality that $\sum_{v \in V} b(v)$ is even. In order to show this fact, let us assume that $\sum_{v \in V} b(v)$ is odd. For such b, any optimal solution $G = (V, E)$ to $(V, r, 0, b, w)$ has a vertex u^* with $d(u^*; G) < b(u^*)$ since $\sum_{v \in V} d(v; G)$ is even. Hence G is also optimal for $(V, r, 0, b', w)$, where $b'(u^*) = b(u^*) - 1$ and $b'(v) = b(v)$ for $v \in V - u^*$. Therefore, any approximation algorithm for instances with even $\sum_{v\in V} b(v)$ can be used to approximate those instances with odd $\sum_{v\in V} b(v)$; Apply the algorithm to at most $|V|$ instances each of which is obtained by decreasing $b(v)$ by 1 for a vertex $v \in V$, and then output the best of the obtained solutions.

Our algorithm for $(V, r \geq 2, 0, b, w)$ consists of the following two phases. The first phase finds a feasible solution G_r to $(V, r \geq 2, 0, +\infty, w)$, i.e., G_r is an r-edge-connected graph. At this point, there may be some vertices v that violate the upper degree constraint (i.e., $d(v; G_r) > b(v)$). Moreover notice that G_r has no cut-edge since $r \geq 2$. For now, let us suppose that $b(v) - d(v; G_r)$ is even for each vertex v with $d(v; G_r) > b(v) = \max_{u\in V-v} r(u, v)$. The second phase reduces the degree of each vertex v with $d(v; G_r)$ to at most $b(v)$. This can be done by computing an r_g-edge-connected g-detachment of G_r for a degree specification g such that, for each $v \in V$, $V_v = \{v_1, \ldots, v_{p_v}\}$, $\rho(v_1) = d(v; G_r) - 2(p_v - 1)$, and $\rho(v_i) = 2$ $(v_i \in V_v - \{v_1\})$, where $p_v = 1 + \max\{0, \lceil (d(v; G_r) - b(v))/2 \rceil\}$. Observe that

$$\rho(v_1) = d(v; G_r) - 2\lceil (d(v; G_r) - b(v))/2 \rceil = \begin{cases} b(v) & \text{if } d(v; G_r) - b(v) \text{ is even,} \\ b(v) - 1 & \text{if } d(v; G_r) - b(v) \text{ is odd,} \end{cases}$$

holds for a vertex v with $d(v; G_r) > b(v)$. Since we are assuming that $b(v) - 1 \geq \max_{u\in V-v} r(u, v)$ if $d(v; G_r) - b(v)$ is odd, it holds that $\rho(v_1) \geq \max_{u\in V-v} r(u, v)$ for every $v \in V$. Let G' be the graph obtained from the g-detachment by neglecting all vertices $v_i \in V_v - \{v_1\}$ $(v \in V)$ (i.e., replacing edges uv_i, v_iu' with uu'), and regard v_1 as v. Then $d(v; G') = \rho(v_1) \leq b(v)$ holds for all $v \in V$. Neglecting some vertices from the g-detachment may create self-loops, which will be simply eliminated whenever created. Although this may further reduce the degree of a vertex v, the resulting graph G' still satisfies the degree constraints since $a = 0$. Hence G' satisfies the degree bounds. On the other hand, we can let G' be r-edge-connected by Corollary 4.4 since $\lambda(u, v; G') \geq r_g(u_1, v_1) = \min\{\lambda(u, v; G), \rho(u_1), \rho(v_1)\} \geq r(u, v)$ for every $u, v \in V$. Therefore we can obtain a feasible solution G'.

If there exists a vertex v such that $d(v; G_r) > b(v) = \max_{u\in V-v} r(u, v)$ and $b(v) - d(v; G_r)$ is odd, then $d(v; G') < \max_{u\in V-v} r(u, v)$, where G' is the graph constructed in the above. In this case, we use the detachment of another graph instead of G_r. Let $U = \{v \in V \mid |b(v) - d(v; G_r)| \text{ is odd}\}$, where $|U|$ is even since $\sum_{v\in V} b(v)$ and $\sum_{v\in V} d(v; G_r)$ are even. Furthermore, compute a minimum cost perfect matching M on U, (i.e., solves $(U, 0, 1, 1, w)$), and adds M to G_r to obtain a graph G'_r. Then $|b(v) - d(v; G'_r)|$ is even for all $v \in V$. Hence the second phase transforms G'_r into a feasible solution G'.

Theorem 5.2. *Let us suppose that we can obtain an α-approximate solution G_r for $(V, r \geq 2, 0, +\infty, w)$. If $d(v; G_r) - b(v)$ is odd for each vertex v with $d(v; G_r) > b(v) = \max_{u\in V-v} r(u, v)$, then the problem $(V, r \geq 2, 0, b, w)$ is approximable within α. Otherwise, the problem $(V, r \geq 2, 0, b, w)$ is approximable within $\alpha + 2/k$ for $k = \min_{u,v\in V} r(u, v)$.*

Proof. We have already seen that our algorithm outputs a feasible solution. The second phase does not increase the edge cost since w is metric. Hence it suffices to show that $w(G_r) \leq \alpha w(G^*)$ and $w(M) \leq 2/k \cdot w(G^*)$, where G^* is an optimal solution for $(V, r \geq 2, 0, b, w)$. An optimal solution for $(V, r \geq 2, 0, +\infty, w)$ has the cost at most $w(G^*)$. This implies that $w(G_r) \leq \alpha w(G^*)$. Hence in the following, we show that $w(M) \leq 2/k \cdot w(G^*)$.

Let $2G^*$ be the graph obtained by duplicating every edge in G^*. Since G^* is r-edge-connected and $r > k$, $2G^*$ is $2k$-edge-connected. It is known that any $2k$-edge-connected graph contains k edge-disjoint spanning trees $\{T_1, \ldots, T_k\}$ [36]. Let $j = \arg\min_{1 \le i \le k} w(T_i)$. Then $w(T_j) \le w(2G^*)/k$. Observe that a spanning tree T_j has $|U|/2$ edge-disjoint paths whose end vertices are U. By shortcutting intermediate vertices in the paths, we can obtain a perfect matching on U whose cost is at most $w(T_j) \le w(2G^*)/k = 2/k \cdot w(G^*)$, as required. □

By Theorem 2.11, we can let $\alpha = 2$ in Theorem 5.2.

5.4 Problem with lower and upper degree bounds

We now consider the problem $(V, r \ge 2, a, b, w)$ with lower and upper degree bounds. In general, self-loops may be created from a loopless graph during the second phase of our algorithm in the previous section. Removing those self-loops may violate the lower degree constraints for some vertices to which the self-loops are incident. Thus our algorithm cannot be applied to this general case. However in this section, we show that the problem $(V, r \ge 2, a, b, w)$ is approximable if an upper bound is uniform, i.e., $b(v) = \ell$, $v \in V$ for some $\ell \in \mathbb{Z}_+$. In what follows, we assume without loss of generality that $a(v) \ge \max_{u \in V - v} r(u, v)$ for all $v \in V$ and $|V| \ge 3$.

Theorem 5.3. *Let us suppose that we can obtain an α-approximate solution $G_{r,a}$ for $(V, r \ge 2, a, +\infty, w)$ and $b(v) = \ell$, $v \in V$ for an $\ell \in \mathbb{Z}_+$. If $d(v; G_{r,a}) - b(v)$ is even for each vertex v with $d(v; G_{r,a}) > b(v) = a(v)$, then the problem $(V, r \ge 2, a, b, w)$ is approximable within α. Otherwise, the problem $(V, r \ge 2, a, b, w)$ is approximable within $\alpha + 2/k$ for $k = \min_{u,v \in V} r(u, v)$.*

Proof. If $d(v; G_r) - b(v)$ is even for each vertex v with $d(v; G_{r,a}) > b(v) = a(v)$, then we transform $G_{r,a}$ so that the degree upper constraints are satisfied as in the algorithm in Section 5.3. By Corollary 4.4, edges incident to $v_i \in V_v - v_1$ in the detachment are not parallel for all $v \in V$ and $2 \le i \le p_v$. This implies that neglecting v_i creates no self-loop. Hence we can obtain a feasible solution. Notice that the optimal cost for $(V, r, a, +\infty, w)$ is at most that of (V, r, a, b, w). Hence the solution obtained by this algorithm is an α-approximate solution for (V, r, a, b, w).

Let us consider the latter case in the following. Let $U = \{v \in V \mid a(v) = \ell \text{ and } |d(v; G_{r,a}) - a(v)| \text{ is odd}\}$. If $|U|$ is odd, let $|U|$ be even by adding a vertex u with $a(u) < \ell$ to U. Such vertex u exists by the following reason; Suppose $a(v) = b(v) = \ell$ for all $v \in V$. If ℓ is even, then $U = \{v \in V \mid d(v) \text{ is odd}\}$, which leads to the contradiction that $\sum_{v \in V} d(v)$ is odd. If ℓ is odd, then $|V|$ must be even since otherwise the problem instance would be infeasible. Because $U = \{v \in V \mid d(v) \text{ is even}\}$ and $|U|$ is odd, the size of $V - U = \{v \in V \mid d(v) \text{ is odd}\}$ is also odd, which leads to the above contradiction again. Hence we can let $|U|$ be even.

Then, compute a minimum cost perfect matching M on U and let $G'_{r,a}$ be the union of $G_{r,a}$ and M. As in the former case, we can transform $G'_{r,a}$ into a feasible solution. Since $w(M)$ is at most $2/k$ times the optimal cost as stated in Theorem 5.2, the resulting feasible solution is an $(\alpha + 2/k)$-approximate solution. □

By the 2-approximability of $(V, r, a, +\infty, w)$ stated in Theorem 5.1, we can let $\alpha = 2$ in Theorem 5.3.

5.5 Problem with exact degrees

This section considers the problem with $a = b$ and $r = k$ with some positive $k \in \mathbb{Z}_+$, i.e., feasible solutions are k-edge-connected perfect b-matchings. In the following, we suppose that $b \geq 2$ unless stated otherwise, and propose an approximation algorithm to (V, k, b, b, w). Concretely, we prove that it is ρ-approximable if $b(v) \geq 2$, $v \in V$, where $\rho = 2.5$ if k is even and $\rho = 2.5 + 1.5/k$ if k is odd. Moreover, we show that this factor can be improved when a degree specification is uniform.

5.5.1 Feasibility

For some degree specification b, there is no perfect b-matching. The following theorem provides a necessary and sufficient condition for a degree specification to admit a perfect b-matching. Note that $b(v)$ can be 1 in this theorem.

Theorem 5.4. *Let V be a vertex set with $|V| \geq 2$ and $b : V \to \mathbb{Z}_+$. Then there exists a perfect b-matching if and only if $\sum_{v \in V} b(v)$ is even and $b(v) \leq \sum_{u \in V - v} b(u)$ for each $v \in V$.*

Proof. The necessity is trivial. We show the sufficiency by constructing a perfect b-matching. We let $V = \{v_1, \ldots, v_n\}$ and $B = \sum_{\ell=1}^{n} b(v_\ell)/2$. For $j = 1, \ldots, B$, we define i_j as the minimum integer such that $\sum_{\ell=1}^{i_j} b(v_\ell) \geq j$, and i'_j as the minimum integer such that $\sum_{\ell=1}^{i'_j} b(v_\ell) \geq B+j$. Notice that $\sum_{\ell=1}^{i_j-1} b(v_\ell) < j$ holds by the definition if $i_j \geq 2$. Then we can see that $i_j \neq i'_j$ since otherwise we would have $b(v_{i_j}) = \sum_{\ell=1}^{i_j} b(v_\ell) - \sum_{\ell=1}^{i_j-1} b(v_\ell) > (B + j) - j = B$ if $i_j \geq 2$ and $b(v_{i_j}) \geq B + j > B$ otherwise, which contradicts to the assumption.

Let $M = \{e_j = v_{i_j} v_{i'_j} \mid j = 1, \ldots, B\}$. Then M contains no loop by $i_j \neq i'_j$. Moreover G_M is a perfect b-matching since $|\{j \mid i_j = \ell \text{ or } i'_j = \ell\}| = b(v_i)$, as required. $\square$

Theorem 5.4 does not mention the edge-connectivity. For existence of connected perfect b-matchings, we additionally need the condition that $\sum_{v \in V} b(v) \geq 2(|V| - 1)$ [27]. This is always satisfied if $b(v) \geq 2$, $v \in V$, which we assumed. For $k \geq 2$, the conditions in Theorem 5.4 and $b(v) \geq k$, $v \in V$ are sufficient for the existence of k-edge-connected perfect b-matchings as our algorithm will construct such b-matchings under the conditions.

5.5.2 Algorithm

Now we describe our algorithm to (V, k, b, b, w). The conditions appeared in Theorem 5.4 and $b(v) \geq k$ for all $v \in V$ can be verified in polynomial time, where they are apparently necessary for an instance to have k-edge-connected perfect b-matchings. Hence our algorithm checks them, and if some of them are violated, it outputs message "INFEASIBLE". In the following, we suppose the existence of perfect b-matchings with $b(v) \geq k$ for all $v \in V$. If $2 \leq |V| \leq 3$, then every perfect b-matching is k-edge-connected because any non-empty vertex set $X \subset V$ is $\{v\}$ or $V - \{v\}$ for some $v \in V$, and then $d(X) = d(v) \geq k$. Hence we can assume without loss of generality that $|V| \geq 4$.

Let M be a minimum cost perfect b-matching, which is computable in polynomial time as mentioned in Section 5.1. In addition, let H be a Hamiltonian cycle spanning V constructed by algorithm CHRISTOFIDES for metric TSP, which is described in Section 2.4.3.

Initialization: After testing the feasibility of a given instance, our algorithm first prepares M and $k' = \lceil k/2 \rceil$ copies $H_1, \ldots, H_{k'}$ of H. Let E denote the union $M \cup H_1 \cup \cdots \cup H_{k'}$ of them. Notice that G_E is $2k'$-edge-connected by the existence of edge-disjoint k' Hamiltonian cycles. We call a vertex v in a handling graph G an *excess vertex* if $d(v; G) > b(v)$ (otherwise a *non-excess vertex*). In G_E, all vertices are excess vertices since $d(v; G_E) = b(v) + 2k'$. In the following steps, the algorithm reduces the degree of excess vertices until no excess vertex exists while generating no loops and keeping k-edge-connectivity (Notice that $k < 2k'$ if k is odd). This is achieved by two phases, Phase 1 and Phase 2, as follows.

Phase 1: In this phase, we modify only edges in M while keeping edges in $H_1, \ldots, H_{k'}$ unchanged. We define the following two operations on an excess vertex $v \in V$.

Operation 1: If v has two incident edges xv and yv in M with $x \neq y$, split $\{xv, yv\}$.

Operation 2: If v has two parallel edges uv in M with $d(u) > b(u)$, remove those edges.

We note that Operation 2 is equivalent to splitting two edges between u and v with removing the generated loop. Phase 1 repeats Operations 1 and 2 until none of them is executable. For avoiding ambiguity, we let M' denote M after executing Phase 1, and M denote the original set in what follows. Moreover, let $E' = M' \cup H_1 \cup \cdots \cup H_{k'}$. Note that $d(v) - b(v)$ is always a non-negative even integer throughout (and after) these operations because $d(v; G_E) - b(v) = 2k'$ and each operation decreases the degree of a vertex by 2. If no excess vertex remains in $G_{E'}$, then we are done. We consider the case in which there remain some excess vertices, and show some properties on M' before describing Phase 2.

Claim 5.1. *Every excess vertex in $G_{E'}$ has at least one incident edge in M' and its neighbors in $G_{M'}$ are unique.*

Proof. Since $d(v; G_{E'}) - b(v)$ is a positive even integer for an excess vertex v in $G_{E'}$, it holds $d(v; G_{M'}) = d(v; G_{E'}) - d(v; G_{H_1 \cup \cdots \cup H_{k'}}) \geq (b(v) + 2) - 2k' > 0$, Hence v has at least one incident edges in M'. If neighbors of v in $G_{M'}$ are not unique, Operation 1 can be applied to v. $\square$

For an excess vertex v in $G_{E'}$, let $n(v)$ denote the unique neighbor of v in $G_{M'}$. If $n(v)$ is also an excess vertex in $G_{E'}$, we call the pair $\{v, n(v)\}$ by a *strict pair.*

Claim 5.2. *Let $\{v, n(v)\}$ be a strict pair. Then $d(v; G_{M'}) = d(n(v); G_{M'}) = 1$, k is odd, and $b(v) = b(n(v)) = k$.*

Proof. By Claim 5.1, $d(v; G_{M'}) = d(n(v); G_{M'})$. If $d(v; G_{M'}) = d(n(v); G_{M'}) > 1$, Operation 2 can be applied to v and $n(v)$, a contradiction. Hence $d(v; G_{M'}) = d(n(v); G_{M'}) = 1$ holds. Let $u \in \{v, n(v)\}$. Then it holds that $d(u; G_{E'}) = d(u; G_{H_1 \cup \cdots \cup H_{k'}}) + d(u; G_{M'}) = 2k' + 1 = 2\lceil k/2 \rceil + 1$. Since $d(u; G_{E'}) - b(u)$ is even, $b(u)$ must be odd. This fact and $d(u, G_{E'}) > b(u) \geq k$ indicate that $b(u) = k$ and k is odd. $\square$

By definition, the existence of excess vertices which are in no strict pairs implies that of some non-excess vertices. Upon completion of Phase 1, let N denote the set of non-excess vertices in $G_{E'}$, and S denote the set of strict pairs in $G_{E'}$. If $N = \emptyset$, all excess vertices are in some strict pairs. By Claim 5.2, k is an odd integer in this case, and furthermore $k \geq 3$ by the assumption that $b(v) \geq 2$, $v \in V$ if $k = 1$. From this fact and $|V| \geq 4$, $N = \emptyset$ implies that at least two strict pairs exist (i.e., $|S| \geq 2$).

Phase 2: Now we describe Phase 2. First, we deal with a special case in which V consists of only two strict pairs.

Claim 5.3. *If V consists of two strict pairs after Phase 1, we can transform $G_{E'}$ into a k-edge-connected perfect b-matching without increasing its cost.*

Proof. Let $V = \{u, v, x, y\}$ and $H = \{uv, vx, xy, yu\}$. Now $E' = M' \cup H_1 \cup \cdots \cup H_{k'}$ $(k \geq 2)$. Then either $M' = \{uv, xy\}$ (or $\{vx, yu\}$) or $M' = \{ux, vy\}$ holds. In both cases, we replace $M' \cup H_1 \cup H_2$ by $E'' = \{uv, vx, xy, yu, ux, vy\}$ (see Fig. 5.2). Then, we can see that $d(v; G_{E''}) = 3$ for all $v \in V$ and $G_{E''}$ is 3-edge-connected. Since $d(v; G_{H_i}) = 2$ for $v \in V, i = 3, \ldots, k'$ and G_{H_i} is 2-edge-connected for $i = 3, \ldots, k'$, it holds that $d(v; G_{E'' \cup H_3 \cup \cdots \cup H_{k'}}) = 3 + 2(k' - 2) = k = b(v)$ for $v \in V$ and the edge-connectivity of $G_{E'' \cup H_3 \cup \cdots \cup H_{k'}}$ is $3 + 2(k' - 2) = k$ (The existence of strict pair implies that k is odd by Claim 5.2.).

Hence it suffices to show that $w(E'') \leq w(M') + w(H_1) + w(H_2)$. If $M' = \{ux, vy\}$ (or $\{vx, yu\}$), then it is obvious since $E'' = M' \cup H_1 \subseteq M' \cup H_1 \cup H_2$. Let us consider the other case, i.e., $M' = \{uv, xy\}$. From $M' \cup H_1 \cup H_2$, remove $\{uv, uv\}$, replace $\{xy, yu\}$ by $\{xu\}$, and replace $\{vx, xy\}$ by $\{vy\}$. Then the edge set becomes E'' without increasing edge cost, as required. $\square$

In the following, we assume that $|S| \geq 3$ when $N = \emptyset$. In this case, Phase 2 modifies only edges in H_i, $i = 1, \ldots, k'$ while keeping the edges in M' unchanged. Let $V(H_i)$ denote the set of vertices spanned by H_i. We define *detaching v from cycle H_i* to be an operation that splits the pair $\{uv, vw\} \subseteq H_i$ of edges incident to v. Note that this decreases $d(v)$ by 2, but H_i remains a cycle on $V(H_i) := V(H_i) - \{v\}$. For each excess vertex v in $G_{E'}$, Phase 2 reduces $d(v)$ to $b(v)$ by detaching v from $(d(v; G_{E'}) - b(v))/2$ cycles in $H_1, \ldots, H_{k'}$. We notice that $(d(v; G_{E'}) - b(v))/2 \leq k'$ by $d(v; G_{E'}) - b(v) \leq d(v; G_E) - b(v) = 2k'$. One important point is to keep $|V(H_i)| \geq 2$ for each $i = 1, \ldots, k'$ during Phase 2. In other words, we always select H_i with $|V(H_i)| \geq 3$ to detach an excess vertex. This is necessary because, if we detach a vertex from H_i with $V(H_i) = 2$, then H_i becomes a loop. In addition, we detach the two excess vertices u and v in a strict pair from different cycles in $H_1, \ldots, H_{k'}$, respectively. This is in order to maintain the k-edge-connectivity of $G_{E'}$ as will be explained below.

Claim 5.4. *It is possible to decrease the degree of each excess vertex v in $G_{E'}$ to $b(v)$ by detaching from some cycles in $H_1, \ldots, H_{k'}$ so that $|V(H_i)|$ remains at least 2 for $i = 1, \ldots, k'$ and the two excess vertices in each strict pair are detached from H_i and H_j with $i \neq j$, respectively.*

Proof. First, let us consider the case of $S \neq \emptyset$. Recall $k \geq 3$ and $k' = \lceil k/2 \rceil \geq 2$ in this case. For each strict pair $\{u, v\} \in S$, we detach u and v from different cycles in $H_1, \ldots, H_{k'}$. On

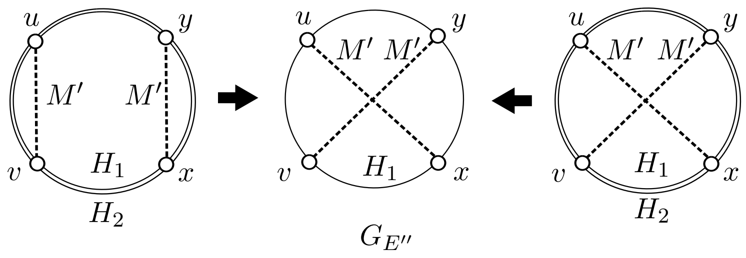


Figure 5.2: Operations when V consists of two strict pairs

the other hand, we detach excess vertex z from arbitrary $(d(z; G_{E'}) - b(z))/2$ cycles. After this, each of $H_1, \ldots, H_{k'}$ is incident to at least one vertex of any strict pair in S in addition to all non-excess vertices in N. By the relation between $|S|$ and $|N|$ we explained in the above, it holds that $|V(H_i)| \geq |S| + |N| \geq 2$ for each $i = 1, \ldots, k'$, as required.

Next, let us consider the case of $S = \emptyset$. As explained in the above, $|N| \geq 1$ holds for this case. If $|N| \geq 2$, the claim is obvious since each of $H_1, \ldots, H_{k'}$ is always incident to all vertices in N. Hence suppose that $|N| = 1$, and let x be the unique non-excess vertex in N. Then all edges in M' are incident to x, since otherwise $S = \emptyset$ implies that Operation 1 or 2 would be applicable to some vertex in $V - x$. In other words, $b(x) = d(x; G_{E'}) = |M'| + 2k'$ holds before Phase 2. Moreover $\sum_{v \in V - x} b(v) \geq b(x)$ also holds by the assumption that perfect b-matchings exist. Now assume that we have converted some excess vertices in $G_{E'}$ into non-excess vertices by detaching them from some of $H_1, \ldots, H_{k'}$ while keeping $|V(H_i)| \geq 2$, $i = 1, \ldots, k'$, and yet an excess vertex $y \in V - x$ remains. Hence $\sum_{v \in V} d(v) > \sum_{v \in V} b(v)$. Then there remains a cycle H_i with $|V(H_i)| > 2$ because

$$
\begin{aligned}
2 \sum_{1 \leq i \leq k'} |V(H_i)| = \sum_{v \in V} d(v; G_{H_1 \cup \cdots \cup H_{k'}}) &= \sum_{v \in V} d(v) - 2|M'| \\
&> \sum_{v \in V - \{x\}} b(v) + b(x) - 2|M'| \geq 2(b(x) - |M'|) \geq 4k'.
\end{aligned}
$$

Therefore we can detach an excess vertex y from such H_i as long as such a vertex exists. This implies that the claim holds also for $|N| = 1$. □

In the following, we let H_i' denote H_i after Phase 2, and H_i denote the original Hamiltonian cycle for $i = 1, \ldots, k'$. Moreover let $E'' = M' \cup H_1' \cup \cdots \cup H_{k'}'$. The algorithm outputs $G_{E''}$. The entire algorithm is described as follows.

Algorithm UNDIRECT(k)

Input: A vertex set V, a degree specification $b : V \to \mathbb{Z}_+$, a metric edge cost $w : V \to \mathbb{Q}_+$, and a positive integer k

Output: A k-edge-connected perfect b-matching or "INFEASIBLE"

1: **if** $\sum_{v \in V} b(v)$ is odd, $\exists v : b(v) > \sum_{u \in V - v} b(u)$ or $k > b(v)$ **then**
2: Output "INFEASIBLE" and halt
3: **end if;**
4: Compute a minimum cost perfect b-matching G_M;

if $|V| \leq 3$ **then**
  Output G_M and halt
end if;
Compute a Hamiltonian cycle G_H on V by algorithm CHRISTOFIDES;
$k' := \lceil k/2 \rceil$; Let $H_1, \ldots, H_{k'}$ be k' copies of H;

\# Phase 1
$M' := M$;
while Operation 1 or 2 is applicable to a vertex $v \in V$
with $d(v; G_{M' \cup H_1 \cup \cdots \cup H_{k'}}) > b(v)$ **do**
  if $\exists \{xv, vy\} \subseteq M'$ such that $x \neq y$ **then**
    $M' := (M' - \{xv, vy\}) \cup \{xy\}$ \# Operation 1
  else
    if $\exists \{xv, vx\} \subseteq M'$ such that $d(x; G_{M' \cup H_1 \cup \cdots \cup H_{k'}}) > b(x)$ **then**
      $M' := M' - \{xv, vx\}$ \# Operation 2
    end if
  end if
end while;

\# Phase 2
if V consists of two strict pairs **then**
  Rename vertices so that $H = \{uv, vx, xy, yu\}$;
  $H'_2 := \emptyset$; $M' := \{ux, vy\}$;
  Output $G_{M' \cup H'_1 \cup \cdots \cup H'_{k'}}$ and halt
end if;
$H'_i := H_i$ for each $i = 1, \ldots, k'$;
while $\exists v \in V$ with $d(v; G_{M' \cup H'_1 \cup \cdots \cup H'_{k'}}) > b(v)$ **do**
  if v and $n(v)$ forms a strict pair **then**
    Detach v from H'_i and $n(v)$ from H'_j, where $i \neq j$
  else
    Detach v from H'_i with $V(H'_i) > 2$
  end if
end while;
$E'' := M' \cup H'_1 \cup \cdots \cup H'_{k'}$;
Output $G_{E''}$

Claim 5.5. *$G_{E''}$ is a k-edge-connected perfect b-matching.*

Proof. We have already seen the case in which V consists of two strict pairs. Hence we suppose the other case in the following. Moreover we have already observed that $d(v; G_{E''}) = b(v)$ holds for each $v \in V$. Furthermore $G_{E''}$ is loopless since G_E is loopless and no operations in the algorithm generate loops. Hence we prove the k-edge-connectivity of $G_{E''}$ below.

Let $u, v \in V$. (i) First suppose that u and v are in some (possibly different) strict pairs in $G_{E'}$. Moreover, let $u \notin V(H'_i)$ and $v \notin V(H'_j)$ (hence $u \in V(H'_{i'})$ for $i' \neq i$ and $v \in V(H'_{j'})$ for $j' \neq j$). For each $\ell \in \{1, \ldots, k'\} - \{i, j\}$, $\lambda(u, v; G_{H'_\ell}) = 2$ holds because $u, v \in V(H'_\ell)$. If

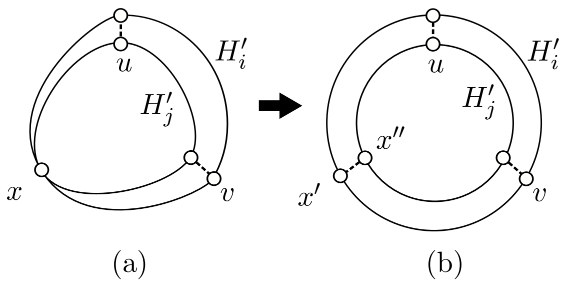


Figure 5.3: Reduction to the case of $V(H'_i) \cap V(H'_j) = \emptyset$

$i = j$, $\lambda(u, v; G_{H'_i \cup M'}) = 1$ holds because $d(u; G_{M'}) = d(v; G_{M'}) = 1$ and $n(u), n(v) \in V(H'_i)$. Then it holds that $\lambda(u, v; G_{E''}) = 2(k' - 1) + 1 = k$ in this case (Recall that the existence of strict pairs implies that k is odd by Claim 5.2). Hence we let $i \neq j$, and show that $\lambda(u, v; G_{H'_i \cup H'_j \cup M'}) \geq 3$ from now on, from which $\lambda(u, v; G_{E''}) \geq 2(k' - 2) + 3 = k$ can be derived.

Let N and S denote the sets of non-excess vertices and strict pairs in $G_{E'}$ after Phase 1, respectively. Suppose that $V(H'_i) \cap V(H'_j) = \emptyset$. In this case, it can be seen that $N = \emptyset$, and hence $|S| \geq 3$ by the assumption about the relation between N and S. Since at least one vertex of each strict pair is spanned by each cycle in $H'_1, \ldots, H'_{k'}$, we can see that M' contains at least three vertex-disjoint edges that join vertices in $V(H'_i)$ and in $V(H'_j)$, two of which are u and v. This indicates that $\lambda(u, v; G_{H'_i \cup H'_j \cup M'}) \geq 3$ holds (see the graph of Figure 5.3 (b)).

Let us consider the case of $V(H'_i) \cap V(H'_j) \neq \emptyset$ in the next. By the existence of u and v, $|S| \geq 1$ holds. If u and v forms a strict pair (i.e., $uv \in M'$), $\lambda(u, v; G_{M'}) = 1$ holds. Since $V(H'_i) \cap V(H'_j) \neq \emptyset$ implies $\lambda(G_{H'_i \cup H'_j}) \geq 2$, we see that $\lambda(u, v; G_{H'_i \cup H'_j \cup M'}) \geq 3$ in this case. Thus let u and v belong to different strict pairs (i.e., $|S| \geq 2$). Then there exists two vertex-disjoint edges in M' joins vertices in $V(H'_i)$ and in $V(H'_j)$ (see Figure 5.3 (a)). If we split each vertex $x \in V(H'_i) \cap V(H'_j)$ into two vertices x' and x'' so that H'_i and H'_j are vertex-disjoint cycles, and add new edges $x'x''$ joining those two split vertices to M', then we can reduce this case to the case of $V(H'_i) \cap V(H'_j) = \emptyset$, in which $\lambda(u, v; G_{H'_i \cup H'_j \cup M'}) \geq 3$ has already been observed in the above (see Figure 5.3). Accordingly, we have $\lambda(u, v; G_{H'_i \cup H'_j \cup M'}) \geq 3$ if u and v are in some strict pairs, as required.

(ii) In the next, let u and v be not in any strict pairs. For $z \in \{u, v\}$, let $n'(z)$ denote z itself if $z \in N$, and $n(z)$ otherwise. Notice that $n'(z) \in N$ for any $z \in \{u, v\}$, i.e., it is spanned by $H'_1, \ldots, H'_{k'}$. If $z \in \{u, v\}$ is not spanned by $p > 0$ cycles in $H'_1, \ldots, H'_{k'}$ (and hence z is an excess vertex in $G_{E'}$), then z has at least $k - 2(k' - p)$ incident edges in M' because $d(z; G_{M'}) = b(z) - d(z; G_{H'_1 \cup \cdots \cup H'_{k'}}) \geq k - 2(k' - p)$. Hence $\lambda(z, n'(z); G_{E''}) \geq 2(k' - p) + k - 2(k' - p) = k$ holds for each $z \in \{u, v\}$, where we define $\lambda(z, z; G_{E''}) = +\infty$. Moreover it is obvious that $\lambda(n'(u), n'(v); G_{E''}) \geq 2k'$. Therefore, it holds that

$$\lambda(u, v; G_{E''}) \geq \min\{\lambda(u, n'(u); G_{E''}), \lambda(n'(u), n'(v); G_{E''}), \lambda(n'(v), v; G_{E''})\} \geq k.$$

(iii) Finally, let us consider the remaining case, i.e., u is in a strict pair and v is a vertex which is not in any strict pair. Let us define $n'(v)$ as in the above. Then $\lambda(v, n'(v); G_{E''}) \geq k$ holds. Without loss of generality, let u be detached from H'_1, and spanned by $H'_2, \ldots, H'_{k'}$.

Since $un(u) \in M'$ and $n(u), n'(v) \in V(H'_1)$, it holds that $\lambda(u, n(u); G_{M' \cup H'_1}) = 1$, and $\lambda(n(u), n'(v); G_{M' \cup H'_1}) \geq 2$. Then,

$$\lambda(u, n'(v); G_{E''}) \geq \min\{\lambda(u, n(u); G_{M' \cup H'_1}), \lambda(n(u), n'(v); G_{M' \cup H'_1})\} + \lambda(u, n'(v); G_{H'_2 \cup \cdots \cup H'_{k'}}) \geq 1 + 2(k'-1) = 2k' - 1 = k.$$

Therefore,

$$\lambda(u, v; G_{E''}) \geq \min\{\lambda(u, n'(v); G_{E''}), \lambda(v, n'(v); G_{E''})\} \geq k,$$

holds, as required. □

Let us consider the cost of the graph $G_{E''}$.

Claim 5.6. *$w(E'')$ is at most $1 + 3\lceil k/2 \rceil / k$ times the optimal cost of (V, k, b, b, w).*

Proof. No operation in Phases 1 and 2 increases the cost of the graph since the edge cost is metric. Hence it suffices to show that $w(M \cup H_1 \cup \cdots \cup H_{k'})$ is at most $(1 + 3\lceil k/2 \rceil / k) \cdot w(E^*)$, where E^* denotes an optimal edge set to (V, k, b, b, w). Since G_{E^*} is a perfect b-matching, $w(M) \leq w(E^*)$ obviously holds. Thus it suffices to show that $w(H_i) \leq 3w(E^*)/k$ for $1 \leq i \leq k'$, from which the claim follows.

Since G_{E^*} is k-edge-connected, $\sum_{e \in \delta(U)} \S_{E^*}(e) \geq k$ holds for every non-empty $U \subset V$. Hence $2\S_{E^*}/k$ is feasible for the linear programming in Theorem 2.10, which means that $\mathrm{OPT}_{TSP} \leq 2w(E^*)/k$. By Theorem 2.10, $w(H_i) \leq 1.5\mathrm{OPT}_{TSP}$. Therefore we have $w(H_i) \leq 3w(E^*)/k$, as required. □

Claims 5.5 and 5.6 establish the next.

Theorem 5.5. *Algorithm UNDIRECT(k) is a ρ-approximation algorithm for (V, k, b, b, w), where $\rho = 2.5$ if k is even and $\rho = 2.5 + 1.5/k$ if k is odd.* □

Algorithm UNDIRECT(k) always outputs a solution for $k \geq 2$ as long as there exists a perfect b-matching and $b(v) \geq k$ for all $v \in V$. This fact and Theorem 5.4 imply the following corollary.

Corollary 5.1. *For $k \geq 2$, there exists a k-edge-connected perfect b-matching if and only if $\sum_{v \in V} b(v)$ is even and $k \leq b(v) \leq \sum_{u \in V - v} b(u)$ for all $v \in V$.* □

One may consider that a perfect $(b - 2k')$-matching is more appropriate than a perfect b-matching as a building block of our algorithm, since there is no excess vertex for the union of a perfect $(b - 2k')$-matching and k' Hamiltonian cycles. However, there is a degree specification b that admits a perfect b-matching, and no perfect $(b - 2k')$-matching. Furthermore, even if there exists a perfect $(b - 2k')$-matching, the minimum cost of the perfect $(b - 2k')$-matching may not be a lower bound on the optimal cost of (V, k, b, b, w). Therefore we do not use a perfect $(b - 2k')$-matching in general case. In Section 5.8, we show that a perfect $(b - 2k')$-matching always exist and its cost can be estimated when a degree specification b is uniform.

5.6 Digraph version of problem with exact degrees

In this section, we extend the result in the previous section to digraphs. A *perfect* (b^-, b^+)-*matching* is a digraph in which in- and out-degrees of each vertex v is $b^-(v)$ and $b^+(v)$, respectively. The problem we consider is described as follows.

k-arc-connected multi-digraph with degree specification (k-ACMDS)
Given a vertex set V, a symmetric metric arc cost $w : V \times V \to \mathbb{Q}_+$, in- and out-degree specifications $b^-, b^+ : V \to \mathbb{Z}_+$, and a positive integer k, find a minimum cost perfect (b^-, b^+)-matching $D = (V, A)$ of arc-connectivity k.

We show that k-ACMDS is 2.5-approximable. Our algorithm for k-ACMDS can be designed analogously with that for (V, k, b, b, w). Before describing the algorithm, we consider the feasibility of k-ACMDS.

5.6.1 Feasibility

Frobenius' classic theorem (see [71] for example) tells the relationship between the existence of perfect bipartite matchings and the minimum size of vertex covers in bipartite graphs.

Theorem 5.6 (Frobenius). *A bipartite graph G has a perfect matching if and only if each vertex cover has size at least $|V(G)|/2$.* □

From this, we can immediately derive a condition for a digraph to have a perfect (b^-, b^+)-matching.

Theorem 5.7. *Let V be a vertex set, and $b^-, b^+ : V \to \mathbb{Z}_+$ be in- and out- degree specifications, respectively. There exists a perfect (b^-, b^+)-matching if and only if $\sum_{v \in V} b^-(v) = \sum_{v \in V} b^+(v)$, $b^-(v) \leq \sum_{u \in V-v} b^+(u)$ for each $v \in V$, and $b^+(v) \leq \sum_{u \in V-v} b^-(u)$ for each $v \in V$.*

Proof. The necessity is obvious. Hence we consider the sufficiency in the following. For each $v \in V$, prepare two vertex sets V_v^- and V_v^+ corresponding to v such that $|V_v^-| = b^-(v)$ and $|V_v^+| = b^+(v)$. Furthermore, let $V^- = \cup_{v \in V} V_v^-$, $V^+ = \cup_{v \in V} V_v^+$, and $E = \{u^- v^+ \mid u^- \in V_u^-, v^+ \in V_v^+, u \neq v\}$. Then a perfect matching in a bipartite graph (V^-, V^+, E) corresponds to a perfect (b^-, b^+)-matching on V. So by Theorem 5.6, it suffices to show that each vertex cover of (V^-, V^+, E) has size at least $(|V^-| + |V^+|)/2$.

To the contrary, let us suppose that there exists a vertex cover $C \subset V^- \cup V^+$ of (V^-, V^+, E) such that $|C| < (|V^-| + |V^+|)/2$ under the assumption in this theorem. Since $|V^-| = \sum_{v \in V} b^-(v) = \sum_{v \in V} b^+(v) = |V^+|$, it holds that $|C| < |V^-| = |V^+|$. This implies the existence of vertices $x \in V^- - C$ and $y \in V^+ - C$. Let x correspond to $u \in V$ (i.e., $x \in V_u^-$) and y correspond to $v \in V$ (i.e., $y \in V_v^+$). If $u \neq v$, there exists an edge $xy \in E$, which is not covered by any vertices in C, a contradiction. Hence $u = v$ holds. Then $\cup_{z \in V-v}(V_z^- \cup V_z^+) \subseteq C$ holds. This implies that $|C| \geq \sum_{z \in V-v} |V_z^-| + \sum_{z \in V-v} |V_z^+|$. Then it holds that

$$\left(\sum_{v\in V} b^-(v) + \sum_{v\in V} b^+(v)\right)/2 = (|V^-| + |V^+|)/2 > |C|$$
$$\geq \sum_{z\in V-v} |V_z^-| + \sum_{z\in V-v} |V_z^+| = \sum_{z\in V-v} b^-(z) + \sum_{z\in V-v} b^+(z),$$

implying $b^-(v) + b^+(v) > \sum_{z\in V-v} b^-(z) + \sum_{z\in V-v} b^+(z)$. However, this indicates that $b^-(v) > \sum_{z\in V-v} b^-(z)$ or $b^+(v) > \sum_{z\in V-v} b^+(z)$ holds, contradicting to the assumption. □

Notice that the proof of Theorem 5.7 indicates the reduction of the minimum cost perfect (b^-, b^+)-matching problem to the minimum cost perfect b-matching problem in an undirected bipartite graph.

5.6.2 Algorithm

We are ready to explain the algorithm for k-ACMDS. In the following, we assume that $b^-(v), b^+(v) \geq k$ for each $v \in V$ and a perfect (b^-, b^+)-matching exists.

Let M be a minimum cost perfect (b^-, b^+)-matching and H be a directed Hamiltonian cycle constructed by the digraph version of algorithm CHRISTOFIDES in Section 2.4.3. for the edge cost obtained from c by ignoring the direction of arcs (Recall that c is symmetric). Moreover let $H_1, \ldots, H_k$ be k copies of H, $A = M \cup H_1 \cup \cdots \cup H_k$, and D_F denote the digraph (V, F) for an arc set F. A vertex $v \in V$ is called an *excess vertex* if $d^-(v) > b^-(v)$ or $d^+(v) > b^+(v)$ (otherwise v is called a *non-excess vertex*). Notice that $d^-(v; D_A) - b^-(v) = d^+(v; D_A) - b^+(v)$. This condition is maintained throughout the algorithm, i.e., $d^-(v) > b^-(v)$ is equivalent to $d^+(v) > b^+(v)$. Our algorithm for k-ACMDS decreases the degree of excess vertices as UNDIRECT(k). One difference between algorithms for (V, k, b, b, c) and for k-ACMDS is the definition of Operations 1 and 2. These will be executed for a pair of arcs entering and leaving the same vertex as follows.

Operation 1: If an excess vertex v has two incident arcs xv and vy in M with $x \neq y$, replace xv and vy by new edge $xy \in M$.

Operation 2: If an excess vertex v has two arcs uv and vu in M with $d^-(u) > b^-(u)$ (and $d^+(v) > b^+(v)$), remove these arcs.

Phase 1 of our algorithm modifies edges in M by repeating Operations 1 and 2 until none of them is executable. We let M' denote M after Phase 1, and M denote the original set in the following. Moreover let $A' = M' \cup H_1 \cup \cdots \cup H_k$, and N denote the set of non-excess vertices in $D_{A'}$. Note that the number of arcs in M' entering (resp., leaving) each excess vertices v in $D_{A'}$ has $d^-(v; D_{A'}) - k \geq d^-(v; D_{A'}) - b^-(v)$ (resp., $d^-(v; D_{A'}) - b^-(v) > d^+(v; D_{A'}) - b^+(v)$) arcs. The other end vertex of them is unique and in N (i.e., a non-excess vertex in $D_{A'}$) since otherwise Operation 1 or 2 can be applied to v. This situation is simpler than after Phase 2 of UNDIRECT(k) since no correspondence of strict pairs exists. Notice that $N \neq \emptyset$ always holds here.

Phase 2 of our algorithm for k-ACMDS modifies edges in $H_1, \ldots, H_k$ so as to decrease the degrees of all excess vertices as in UNDIRECT(k). We repeat *detaching* each excess vertex from some of $H_1, \ldots, H_k$, where detaching a vertex v from H_i is defined as an operation that replaces the pair $\{uv, vw\} \subseteq H_i$ of arcs entering and leaving v by new arc uw. We can prove that it is possible to detach excess vertices from Hamiltonian cycles while keeping $V(H_i) \geq 2$ for $1 \leq i \leq k$ as in UNDIRECT(k).

Claim 5.7. *It is possible to decrease the degree of each excess vertex v to $b(v)$ by detaching v from some cycles in $H_1, \ldots, H_k$ so that $|V(H_i)|$ remains at least two for all $i = 1, \ldots, k$.*

Proof. Recall that $N \neq \emptyset$. If $|N| \geq 2$, the claim is obvious since each of $H_1, \cdots, H_k$ is incident to all vertices in N. Hence suppose that $|N| = 1$, and let x be the unique vertex in N. Then all arcs in M' are incident to x since otherwise Operation 1 or 2 would be applicable to some vertex in $V - x$. In other words, it hold $|M'| = d^-(x; D_{M'}) + d^+(v; D_{M'}) = b^-(x) + b^+(x) - 2k$. Recall that $\sum_{v \in V-x} b^+(v) \geq b^-(x)$ and $\sum_{v \in V-x} b^-(v) \geq b^+(x)$ hold by the assumption that perfect (b^-, b^+)-matchings exist. Now assume that we have converted some excess vertices in $D_{A'}$ into non-excess vertices by detaching them from some of $H_1, \ldots, H_k$ while keeping $|V(H_i)| \geq 2$, $i = 1, \ldots, k$, and yet an excess vertex $y \in V - x$ remains. Then there remains a cycles H_i with $|V(H_i)| > 2$ because

$$\begin{aligned}\sum_{1 \leq i \leq k} |V(H_i)| = \sum_{v \in V} d^-(v; D_{H_1 \cup \cdots \cup H_k}) &= \sum_{v \in V} d^-(v; D_{E'}) - |M'| \\ &> \sum_{v \in V - \{x\}} b^-(v) + d^-(x; D_{E'}) - |M'| \geq b^+(x) + b^-(x) - |M'| \geq 2k.\end{aligned}$$

Hence we can detach y from such H_i, implying the claim also for $|N| = 1$. □

In the following, we let H'_i denote H_i after Phase 2, and H_i denote the original Hamiltonian cycle for $i = 1, \ldots, k$ in order to avoid the ambiguity. Moreover let $A'' = M' \cup H'_1 \cup \cdots \cup H'_k$. Our algorithm outputs $D_{A''}$ as a solution.

Algorithm DIRECT(k)

Input: A vertex set V, in- and out-degree specification $b^-, b^+ : V \to \mathbb{Z}_+$, a symmetric metric arc cost $w : V \times V \to \mathbb{Q}_+$, and a positive integer k

Output: A k-arc-connected perfect (b^-, b^+)-matching or "INFEASIBLE"

if $\sum_{v \in V} b^-(v) \neq \sum_{v \in V} b^+(v)$, $\exists v : b^-(v) > \sum_{u \in V-v} b^+(u)$, $\exists v : b^+(v) > \sum_{u \in V-v} b^-(u)$, $\exists v : k > b^-(v)$, or $\exists v : k > b^+(v)$ **then**
  Output "INFEASIBLE" and halt
end if;
Compute a minimum cost perfect (b^-, b^+)-matching D_M;
Compute a Hamiltonian cycle D_H on V by the digraph version of algorithm CHRISTOFIDES; Let $H_1, \ldots, H_k$ be k copies of H;

\# Phase 1
$M' := M$;
while Operation 1 or 2 is applicable to a vertex $v \in V$ with $d^-(v; D_{M' \cup H_1 \cup \cdots \cup H_k}) > b^-(v)$ **do**
  if $\exists \{xv, vy\} \subseteq M'$ such that $x \neq y$ **then**

$M' := (M' - \{xv, vy\}) \cup \{xy\}$ # Operation 1
else if $\exists \{xv, vx\} \subseteq M'$ such that $d^-(x; D_{M' \cup H_1 \cup \cdots \cup H_k}) > b^-(x)$ **then**
$M' := M' - \{xv, vx\}$ # Operation 2
end if
end while;

Phase 2
$H'_i := H_i$ for each $i = 1, \ldots, k$;
while $\exists v \in V$ with $d^-(v; D_{M' \cup H'_1 \cup \cdots \cup H'_k}) > b^-(v)$ **do**
Detach v from H'_i with $V(H'_i) > 2$
end while;
$A'' := M' \cup H'_1 \cup \cdots \cup H'_k$;
Output $D_{A''}$

Algorithm DIRECT(k) outputs a feasible solution.

Claim 5.8. *$D_{A''}$ is a k-arc-connected perfect (b^-, b^+)-matching.*

Proof. By Claim , $D_{A''}$ is a perfect (b^-, b^+)-matching. Hence in the following, we show that $D_{A''}$ is k-arc-connected.

Let us consider a pair $\{u, v\}$ of two vertices in V. For a vertex $z \in \{u, v\}$, let $n'(z)$ denote z itself if $z \in N$, and the unique neighbor of z in $G_{M'}$ otherwise. Notice that $n'(z) \in N$ for any $z \in \{u, v\}$, i.e., it is spanned by $H'_1, \ldots, H'_k$. If $z \in \{u, v\}$ is not spanned by $p > 0$ cycles in $H'_1, \ldots, H'_k$ (and hence z is an excess vertex in $D_{A'}$), then z has at least p arcs leaving z and p arcs entering z in M' because $d^-(z; D_{M'}) = b^-(z) - d^-(z; D_{H'_1 \cup \cdots \cup H'_k}) \geq k - (k-p) = p$ and $d^+(z; D_{M'}) = b^+(z) - d^+(z; D_{H'_1 \cup \cdots \cup H'_k}) \geq k - (k-p) = p$. Hence $\lambda(z, n'(z); D_{A''}) \geq (k-p) + p = k$ and $\lambda(n'(z), z; D_{A''}) \geq (k-p) + p = k$ hold for each $z \in \{u, v\}$, where we define $\lambda(z, z; D_{A''}) = +\infty$. Moreover it is obvious that $\lambda(n'(u), n'(v); D_{A''}) \geq k$. Therefore, it holds that

$$\lambda(u, v; D_{A''}) \geq \min\{\lambda(u, n'(u); D_{A''}), \lambda(n'(u), n'(v); D_{A''}), \lambda(n'(v), v; D_{A''})\} \geq k.$$

□

The following estimates the cost of solution output by algorithm DIRECT(k).

Claim 5.9. *$w(A'')$ is at most 2.5 times the optimal cost of k-ACMDS.*

Proof. Let A^* be an optimal arc set of k-ACMDS. It is obvious that $w(M) \leq w(A^*)$ since A^* is a perfect (b^-, b^+)-matching. In the following, we show $w(H_i) \leq 1.5w(A^*)/k$ for $1 \leq i \leq k$.

For an arc set A, let E_A denote the set of undirected edges obtained by ignoring the direction of every arc in A. Moreover define a metric edge cost $w' : \binom{V}{2} \to \mathbb{Q}_+$ such that $w'(E_e) = w(e)$ for each $e \in \binom{V}{2}$. Since G_{E_A} is $2k$-edge-connected, $\sum_{e \in \delta(U)} \S_{E_A}(e) \geq 2k$ holds for every non-empty $U \subset V$. Hence $\S_{E_A}/k$ is feasible for the linear programming in Theorem 2.10, which means that $\mathrm{OPT}_{TSP} \leq w'(E_A)/k$. We note that algorithm CHRISTOFIDES outputs E_{H_i} to V and w'. Therefore by Theorem 2.10, $w'(E_{H_i}) \leq 1.5\mathrm{OPT}_{TSP}$ holds. Accordingly we have $w(H_i) = w'(E_{H_i}) \leq 1.5w'(E_A)/k = 1.5w(A)/k$, as required. □

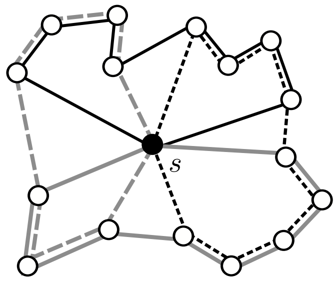


Figure 5.4: A feasible solution for $(4,2)$-VRP

Claims 5.8 and 5.9 establishes the next.

Theorem 5.8. *Algorithm DIRECT(k) is a 2.5-approximation algorithm for k-ACMDS.* □

Algorithm DIRECT(k) always outputs a solution when there exists a perfect (b^-,b^+)-matching and $b^-(v) \geq k$, $b^+(v) \geq k$ for all $v \in V$. This fact and Theorem 5.7 implies the following corollary.

Corollary 5.2. *For $k \geq 1$, there exists a k-arc-connected perfect (b^-,b^+)-matching if and only if $\sum_{v\in V} b^-(v) = \sum_{v\in V} b^+(v)$, $k \leq b^-(v) \leq \sum_{u\in V-v} b^+(u)$ for each $v \in V$, and $k \leq b^+(v) \leq \sum_{u\in V-v} b^-(u)$ for each $v \in V$.* □

5.7 Generalizing VRP

In this section, we consider the following generalization of VRP.

(m,n)**-VRP**
Given a vertex set V containing a special vertex s, a metric edge cost $w : \binom{V}{2} \to \mathbb{Q}_+$, and non-negative integers m and n, find a minimum cost set of m cycles, each containing s, such that each vertex in $V - s$ is contained in exactly n of those cycles.

Figure 5.4 illustrates a feasible solution for $(4,2)$-VRP.

We can assume without loss of generality that $n \leq m \leq n(|V|-1)$ since otherwise the instance is clearly infeasible. An example of applying the (m,n)-VRP is the schedule of garbage collection. Let us consider the case in which a garbage collecting truck must visit each city on n of 5 weekdays in a week. A solution of $(5,n)$-VRP gives a schedule of this truck minimizing total length of routes.

Each solution to (m,n)-VRP is obviously feasible to $(V,2n,b,b,w)$ with $b(s) = 2m$ and $b(v) = 2n$ for $v \in V - s$ (Hence the optimal value of $(V,2n,b,b,w)$ with such b is at most that of (m,n)-VRP). However, the opposite direction does not hold as an example in Figure 5.7. Nevertheless we can see that algorithm UNDIRECT$(2n)$ outputs a feasible solution for (m,n)-VRP.

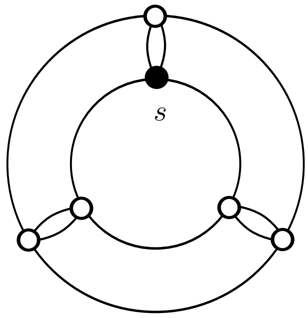


Figure 5.5: A solution to $(V, 4, 4, 4, c)$ that is not feasible to (2, 2)-VRP

Theorem 5.9. *Let $b(s) = 2m$, $b(v) = 2n$ for each $v \in V - s$ and $k = 2n$. Then algorithm UNDIRECT(k) outputs a 2.5-approximate solution to (m, n)-VRP.*

Proof. The solution given by algorithm UNDIRECT$(2n)$ consists of edge set M' and cycles $H'_1, \ldots, H'_n$. In what follows, we see that this solution is feasible to (m, n)-VRP.

Let us consider the moment after Phase 1, and define E', M' and $H'_1, \ldots, H'_{k'}$ as in Section 5.5. Since $k = 2n$ is even, there exists no strict pair. Hence at least one end vertex of each edge in M' is a non-excess vertex. Let v be such a vertex. Then $b(v) = d(v; G_{E'}) > d(v; G_{H_1 \cup \cdots \cup H_n}) = 2n$ (Recall that each non-excess vertex is covered by all of $H_1, \ldots, H_n$). However, a vertex of degree more than $2n$ is only s since $b(u) = 2n$ for each $u \in V - s$. Hence we can see that (i) s is a non-excess vertex after Phase 1, and (ii) one end vertex of each in M' is s. Condition (i) implies that each of $H'_1, \ldots, H'_n$ covers s. Condition (ii) indicates that edges between s and a vertex $v \in V - s$ forms $d(v; M')/2$ cycles whose vertex sets are $\{s, v\}$ because $d(v; M')$ is even. Therefore, combining the fact that $d(v; G_{M' \cup H'_1 \cup \cdots \cup H'_n}) = b(v)$ for all $v \in V$, these show that UNDIRECT(k) outputs a feasible solution to (m, n)-VRP. □

The approximation factor can be improved as follows.

Theorem 5.10. *Problem (m, n)-VRP can be approximated within a factor of* $1.5 + (m - n)/m$.

Proof. Let $b(s) = 2m$, $b(v) = 2n$ for each $v \in V - s$ and $k = 2n$. Moreover, let E be an optimal solution for (m, n)-VRP, and F be the set of edges contained by $m - n$ cycles in G_E of least cost. Then it holds that $d(s; G_F) = 2m - 2n$ and $d(v; G_F) \leq 2n$ for $v \in V - s$. Besides this, we have $w(F) \leq \frac{m-n}{m} w(E)$ by the definition of F.

Now we let $V - s = \{v_1, \ldots, v_{|V|-1}\}$ so that $w(sv_1) \leq w(sv_2) \leq \cdots \leq w(sv_{|V|-1})$. Moreover we define R as an edge set which consists of $2n$ edges sv_i for each $i = 1, \ldots, p$ and $2m - 2n(p + 1)$ edges sv_{p+1}, where $p = \lfloor (m - n)/n \rfloor$. Then it is clear that R is a minimum cost edge set such that $d(s; G_R) = 2np + 2m - 2n(p + 1) = 2m - 2n$ and $d(v; G_R) \leq 2n$ for all $v \in V - s$. This implies that $c(R) \leq w(F) \leq \frac{m-n}{m} w(E)$.

By using R instead of M in UNDIRECT(k), we can obtain a feasible solution to (V, k, b, b, w). As in Theorem 5.9, this solution is also feasible to (m, n)-VRP. Moreover the cost of the solution is at most $w(H_1) + \cdots + w(H_{k'}) + w(R) \leq (1.5 + \frac{m-n}{m}) w(E)$, which completes the proof. □

5.8 Uniform degree specification

Let ℓ be some positive integer at most k. In this section, we show that the approximation factor of our algorithms can be improved for (V, k, ℓ, ℓ, w) and for k-ACMDS with $b^- = b^+ = \ell$. We call a perfect b-matching (resp., a perfect (b^-, b^+)-matching) M *ℓ-regular* if $b = \ell$ (resp., $b^- = b^+ = \ell$).

Lemma 5.1. *Assume that an ℓ-regular digraph exists. Let OPT denote the optimal cost of k-ACMDS with $b^- = b^+ = \ell$. Then there exists an $(\ell - m)$-regular digraph D_R with $w(R) \leq (\ell - m)\mathrm{OPT}/\ell$ for an arbitrary non-negative integer $m \leq \ell$.*

Proof. Let A denote an optimal arc set of k-ACMDS. As seen in Section 5.6, digraph D_A corresponds to the bipartite undirected graph (V^-, V^+, E), which is ℓ-regular. A theorem derived from Frobenius' theorem tells that every ℓ-regular bipartite graph can be decomposed into ℓ graphs each of which is 1-regular [71]. Let R be the set of arcs corresponding to edges in least cost $\ell - m$ graphs of them. Then R is $(\ell - m)$-regular and $w(R) \leq \frac{\ell - m}{\ell} w(A)$, as required. □

The union of an $(\ell - k)$-regular digraph and k Hamiltonian cycles are obviously feasible to k-ACMDS with $b^- = b^+ = \ell$. Therefore we can derive the following theorem.

Theorem 5.11. *k-ACMDS with $b^- = b^+ = \ell$ is approximable within a factor of $1.5 + (\ell - k)/\ell$.* □

Next, we consider (V, k, ℓ, ℓ, w).

Lemma 5.2. *Assume that an ℓ-regular graph exists. Let OPT denote the optimal cost of (V, k, ℓ, ℓ, w). Then there exists an $(\ell - 2m)$-regular graph G_R such that $w(R) \leq \frac{\ell - 2m}{\ell}\mathrm{OPT}$ if ℓ is even, and $w(R) \leq (\frac{\ell - 2m - 1}{\ell} + \frac{1}{k})\mathrm{OPT}$ if ℓ is odd for an arbitrary non-negative integer m with $2m \leq \ell$.*

Proof. Let E denote an optimal edge set of (V, k, ℓ, ℓ, w). First suppose that ℓ is even. Then E can be oriented into an arc set A such that D_A is $\ell/2$-regular. Let w' be an arc cost on A defined naturally from w (i.e., $w'(a) = w(e)$ if $a \in A$ corresponds to $e \in E$). As in the proof of Lemma 5.1, we can obtain an $(\ell/2 - m)$-regular digraph R' with $w'(R') \leq \frac{\ell/2 - m}{\ell/2} w'(A)$. Let R be an edge set corresponding to R'. Then clearly G_R is $(\ell - 2m)$-regular and $w(R) \leq \frac{\ell/2 - m}{\ell/2} w(E)$, as required.

Next, suppose that ℓ is odd. Let $2E$ denote the edge set obtained by duplicating each edge in E. Then G_{2E} is 2ℓ-regular. By the above argument about the case of ℓ is even, we can obtain an $(\ell - 2m - 1)$-regular graph G_F such that $w(F) \leq \frac{\ell - 2m - 1}{2\ell} w(2E) = \frac{\ell - 2m - 1}{\ell} w(E)$ (Notice that $\ell - 2m - 1$ is even). Let M be a minimum cost 1-regular graph. Notice that such M exists since $|V|$ is even by the existence of an ℓ-regular graph with odd ℓ. Since the minimum cost of Hamiltonian cycles spanning all vertices is at most $2w(E)/k$ as shown in the proof of Claim 5.6, we can see that $w(M) \leq w(E)/k$. Let $R = F \cup M$. Then G_R is $(\ell - 2m)$-regular and $w(R) = w(F) + w(M) \leq (\frac{\ell - 2m - 1}{\ell} + \frac{1}{k}) w(E)$, as required. □

Let $k' = \lceil k/2 \rceil$. The union of an $(\ell - 2k')$-regular graph and $2k'$ Hamiltonian cycles are obviously feasible to (V, k, ℓ, ℓ, c). Therefore we can derive the following theorem.

Theorem 5.12. *Problem (V, k, ℓ, ℓ, w) is approximable within a factor of $\frac{\ell-2k'}{\ell} + 3\frac{k'}{k}$ if ℓ is even, and $\frac{(\ell-2k'-1)}{\ell} + \frac{1+3k'}{k}$ if ℓ is odd, where $k' = \lceil k/2 \rceil$.* $\square$

Recall that metric TSP can be formulated as $(V, 2, 2, 2, w)$. Theorem 5.12 indicates that this case can be approximated within 1.5 as algorithm CHRISTOFIDES.

Chapter 6

The Set Connector Problem

This chapter introduces a set connector problem which is defined on an edge-connectivity among vertex subsets. We derive an approximate integer decomposition property from a fractional packing theorem, and present an approximation algorithm for the set connector problem.

6.1 Introduction

For a family $\mathcal{V} \subseteq 2^V$ of disjoint vertex subsets, we let $G/\mathcal{V}$ stand for the graph obtained from G by contracting each $X \in \mathcal{V}$ into a single vertex x, which is called a *$\mathcal{V}$-terminal*. As a general concept of the edge connectivity between two vertices, we define the edge-connectivity $\lambda(\mathcal{V}; G)$ for $\mathcal{V} \subseteq 2^V$ as the minimum of the maximum number of edge-disjoint paths between two $\mathcal{V}$-terminals in $G/\mathcal{V}$. If $\mathcal{V}$ consists of two singletons $\{u\}$ and $\{v\}$, then $\lambda(\mathcal{V}; G)$ is equivalent to the local edge-connectivity between two vertices u and v. By Menger's theorem, we can see that $\lambda(\mathcal{V}; G) = \min\{|\delta(X)| \mid X \subset V \text{ separates } \mathcal{V}\}$ holds.

In this chapter, we consider the *set connector problem*, which is defined as follows.

> **Set connector problem**
> Given a simple undirected graph $G = (V, E)$, an edge cost $w : E \to \mathbb{Q}_+$, and families $\mathcal{V}_1, \ldots, \mathcal{V}_m \subseteq 2^V$ of disjoint vertex subsets, find a minimum cost edge subset $F \subseteq E$ such that $\lambda(\mathcal{V}_i; G_F) \geq 1$ for $1 \leq i \leq m$.

We call a feasible solution for the set connector problem a *set connector*. Notice that a minimal set connector is a forest. Figure 6.1 shows an instance $(G, \mathcal{V}_1 = \{U, W, Z\}, \mathcal{V}_2 = \{U, X, Y\})$ of the set connector problem, where the subsets $U, W, X, Y, Z \subseteq V$ are respectively depicted by gray areas, and a set connector F is given by the edges represented by dashed lines.

The set connector problem contains many fundamental problems. For example, it is equivalent to the Steiner forest problem when each $\mathcal{V}_i$ consists of two singletons. Besides this, it contains the group Steiner tree problem, which is another generalization of the Steiner tree problem. As will be observed in Section 6.5, the group Steiner tree problem contains the set cover problem, the tree cover problem, and the terminal Steiner tree problem as its special cases.

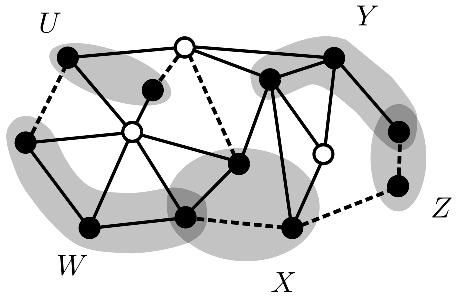


Figure 6.1: An instance of the set connector problem with $\mathcal{V}_1 = \{U, W, Z\}$ and $\mathcal{V}_2 = \{U, X, Y\}$, where a set connector F consists of the edges depicted by dashed lines.

Our main contribution to this problem is to present a 2α-approximation algorithm, where $\alpha = \max_{1\le i\le m}(\sum_{X\in\mathcal{V}_i}|X|) - 1$. To the best of our knowledge, this is the first approximation algorithm that approximates the Steiner forest problem and the group Steiner tree problem simultaneously. The approximation ratio of our algorithm to the set connector problem matches with the best approximation ratios of several special cases such as the Steiner forest problem, as will be discussed in Section 6.5. Our algorithm is based on the *approximate integer decomposition property* [11]. A polyhedron P has an f-approximate integer decomposition property for a real $f > 0$ if, for every rational vector $x \in$ P, there exist k integer vectors $x_1, \dots, x_k \in$ P such that kx is an integer vector and $fkx \ge x_1 + \cdots + x_k$ holds. This property implies that the integrality gap of polyhedron P is at most f, since an integer vector x_j attaining $\min\{w^T x_i \mid i = 1, \dots, k\}$ satisfies $fw^T x \ge w^T x_j$ for any cost vector w. C. Chekuri and F. B. Shepherd [11] showed the 2-approximate integer decomposition property of an LP relaxation for the Steiner forest problem via the following Steiner packing theorem, which generalizes a well-known spanning tree packing theorem due to Gusfield [36] to the case of Eulerian graphs.

Theorem 6.1 ([11]). *Let G be an Eulerian multigraph. Then G contains k edge-disjoint forests $F_1, \dots, F_k$ such that $\lambda(u, v; G_{F_i}) \ge 1$, $1 \le i \le k$ holds for every two vertices u and v that belong to the same $2k$-edge-connected component in G.* □

The set connector problem can be formulated as the following integer programming.

$$\begin{array}{lll} \text{minimize} & w^T x & \\ \text{subject to} & x(\delta(X)) \ge 1 & \text{for every } X \subset V \text{ separating some } \mathcal{V}_i \in \{\mathcal{V}_1, \dots, \mathcal{V}_m\} \\ & x \in \{0,1\}^E. & \end{array}$$

Let $\mathrm{LP_{sc}}$ be the linear programming obtained by relaxing the integrality constraint $x \in \{0,1\}^E$ of this problem into $x \in \mathbb{R}_+^E$, and let $\mathrm{P_{sc}}$ denote its feasible region. For obtaining the 2α-approximate integer decomposition property of $\mathrm{P_{sc}}$, it suffices to show the following set connector packing theorem, which is a generalization of Theorem 6.1.

Theorem 6.2. *Let G be an Eulerian multigraph, $\mathcal{V}_1, \dots, \mathcal{V}_m$ be families of disjoint vertex subsets, and $\alpha = \max_{1\le i\le m}(\sum_{X\in\mathcal{V}_i}|X|) - 1$. If $\lambda(\mathcal{V}_i; G) \ge 2\alpha k$ for $1 \le i \le m$, then G contains k edge-disjoint set connectors.* □

The approximate integer decomposition property depends on the fact that $x \in \mathrm{P}$ is a rational vector. Hence we actually prove the following fractional packing theorem instead of Theorem 6.2. The proof of the theorem can be easily modified to imply Theorem 6.2.

Theorem 6.3. *Let $x \in \mathrm{P}_{\mathrm{sc}}$ and $\alpha = \max_{1\leq i\leq m}(\sum_{X\in\mathcal{V}_i}|X|)-1$ for a simple undirected graph $G=(V,E)$ and families $\mathcal{V}_1,\dots,\mathcal{V}_m$ of disjoint vertex subsets. Then there exist set connectors $C_1,\dots,C_k$ with positive weights $\mu_1,\dots,\mu_k$ such that $2\alpha x \geq \sum_{i=1}^{k}\mu_i\mathcal{X}_{C_i}$ and $\sum_{i=1}^{k}\mu_i = 1$, where $\mathcal{X}_{C_i} \in \{0,1\}^E$ denotes the incidence vector of C_i.* $\square$

We denote a weighted subgraph (V,F), $F \subseteq \binom{V}{2}$ by (F,μ), where μ is a positive real. A set of weighted subgraphs $(F_1,\mu_1),\dots,(F_k,\mu_k)$ is called a *fractional forest packing* of an edge-weighted graph (V,x) if F_i is a forest, $1\leq i\leq k$, $x \geq \sum_{1\leq i\leq k}\mu_i\mathcal{X}_{F_i}$, and $\sum_{1\leq i\leq k}\mu_i = 1$. Notice that $F_i \subseteq E_x$ holds for $1\leq i\leq k$ here. If each of $F_1,\dots,F_k$ is a spanning tree on V (resp., set connector), we especially call it *fractional spanning tree packing* (resp., *fractional set connector packing*). We may simply say that a set of edge subsets $F_1,\dots,F_k$ is a fractional forest packing of (V,x) if there are weights $\mu_1,\dots,\mu_k$ such that $\{(F_i,\mu_i) \mid i=1,\dots,k\}$ is a fractional forest packing of (V,x).

6.2 Contraction and splitting

In this section, we review graph operations called *contraction* and *splitting*. We use these operations in order to prove some claims inductively in subsequent sections.

Contracting a vertex set $S \subseteq V$ into a single vertex s means that S is replaced by s, resultant loops are deleted, and one end vertex of every edge in $\delta(S)$ is changed from a vertex in S to s. Let V' denote the vertex set obtained by the contraction, i.e., $V' = (V-S)\cup s$. If we execute the contraction in (V,x), then x is modified into $x' \in \mathbb{R}^{\binom{V'}{2}}$ so that $x'(e) = x(e)$ for each $e \in \binom{V'}{2} - \delta(s)$, and $x'(e) = \sum_{u\in S}x(uv)$ for each $e = sv \in \delta(s)$.

Lemma 6.1. *Let (V',x') be an edge-weighted undirected graph obtained from (V,x) by contracting $S \subseteq V$ into a single vertex s. If there exists a fractional forest packing $\mathcal{C}$ of (V',x'), then we can obtain a fractional forest packing $\mathcal{C}'$ of (V,x) in which every forest consists of edges in $E_x - \binom{S}{2}$. Every two vertices in $V-S$ connected by all forests in $\mathcal{C}$ are also connected by the union of every forest in $\mathcal{C}'$ and every spanning tree on S.*

Proof. Let $(F,\mu) \in \mathcal{C}$ and $F \cap \delta(s) = \{sv_1,\dots,sv_{d(s;G_F)}\}$. For an ordered multiset $u = (u_1,\dots,u_{d(s;G_F)}) \in S^{d(s;G_F)}$, we define

$$F_u = (F-\delta(s)) \cup \{u_1v_1,\dots,u_{d(s;G_F)}v_{d(s;G_F)}\}$$

and

$$\mu_u = \mu \prod_{i=1}^{d(s;G_F)} \frac{x(u_iv_i)}{x'(sv_i)}.$$

In what follows, we observe that

$$\mathcal{C}' = \{(F_u,\mu_u) \mid (F,\mu)\in\mathcal{C}, u=(u_1,\dots,u_{d(s;G_F)}) \in S^{d(s;G_F)} \text{ and } x(u_iv_i)>0 \text{ for } i=1,\dots,d(s;G_F)\}$$

gives a required fractional forest packing.

By the definition, a forest F_u in $\mathcal{C}'$ is a subset of $E_x - \binom{S}{2}$, and the union of F_u and a tree spanning S connects every two vertices connected by F. Moreover, verify that

$$\sum_{(F,\mu)\in\mathcal{C}'} \mu = \sum_{(F,\mu)\in\mathcal{C}} \sum_{u\in S^{d(s;G_F)}} \mu_u = \sum_{(F,\mu)\in\mathcal{C}} \mu = 1$$

holds. In the last, specify an arbitrary $v \in V - S$, and let $\mathcal{L} = \{(F,\mu) \in \mathcal{C} \mid sv \in F\}$. Now we can suppose without loss of generality that $F \cap \delta(s) = \{sv\}$ for every forest F in $\mathcal{L}$. For every $u \in S$,

$$\sum_{(F,\mu)\in\mathcal{C}'} \mu\mathcal{X}_F(uv) = \sum_{(F,\mu)\in\mathcal{L}} \mu_u = \sum_{(F,\mu)\in\mathcal{L}} \mu\frac{x(uv)}{x'(sv)} \le x(uv)$$

holds because $\sum_{(F,\mu)\in L} \mu \le x'(sv)$. Combining this and the fact that $\sum_{(F,\mu)\in\mathcal{C}'} \mu\mathcal{X}_F(e) \le x'(e) = x(e)$ for $e \in \binom{V-S}{2}$ show that $\sum_{(F,\mu)\in\mathcal{C}'} \mu\mathcal{X}_F \le x$. Accordingly the proof is completed. □

We have already seen the definitions of splitting and complete splitting in Section 2.5.2. Recall that the complete splitting can be executed in strongly polynomial time. Splitting can be also used for the induction as described below.

Lemma 6.2. *Let $x' \in \mathbb{R}^{\binom{V-s}{2}}$ be the edge-weight obtained from $x \in \mathbb{R}^{\binom{V}{2}}$ by a complete splitting at s. If there exists a fractional forest packing $\mathcal{C}$ of (V,x'), then we can construct a fractional forest packing $\mathcal{C}'$ of (V,x). Two vertices in $V - s$ connected by each forest in $\mathcal{C}$ are also connected by each in $\mathcal{C}'$.*

Proof. In the following, we describe how to construct $\mathcal{C}'$ from $\mathcal{C}$. We suppose that the complete splitting consists of only splitting $\{sa, sb\}$ by $\epsilon > 0$ (i.e., $x(sa) = x(sb) = \epsilon$ and $x(sv) = 0$ for $v \in V - \{s,a,b\}$) since otherwise it suffices to repeat the procedure described below.

Let $\mathcal{L} \subseteq \mathcal{C}$ denote the set of weighted forests containing the edge ab. Define $F' = (F - ab) \cup \{sa, sb\}$ and $\mu' = \epsilon\mu/x'(ab)$ for each $(F,\mu) \in \mathcal{L}$. Observe that F' connects every two vertices in $V - s$ that are connected by F. In addition, prepare a new weight $\mu'' = x(ab)\mu_i/x'(ab)$ for F. Then $\mathcal{C}' = (\mathcal{C} - \mathcal{L}) \cup \{(F',\mu'), (F,\mu'') \mid (F,\mu) \in \mathcal{L}\}$ is a required fractional forest packing because of the following facts.

Firstly, it holds that

$$\sum_{(F,\mu)\in\mathcal{C}'} \mu\mathcal{X}_F(ab) = \sum_{(F,\mu)\in\mathcal{L}} \mu'' \le x(ab).$$

Secondly,

$$\sum_{(F,\mu)\in\mathcal{C}'} \mu\mathcal{X}_F(e) = \sum_{(F,\mu)\in\mathcal{L}} \mu' \le \epsilon$$

holds for each $e \in \{sa, sb\}$. Finally,

$$\sum_{(F,\mu)\in\mathcal{C}'} \mu\mathcal{X}_F(e) = \sum_{(F,\mu)\in\mathcal{C}} \mu\mathcal{X}_F(e) \le x'(e) = x(e)$$

holds for each $e \in \binom{V}{2} - \{sa, sb, ab\}$. Therefore, we have $\sum_{(F,\mu)\in\mathcal{C}'} \mu\mathcal{X}_F \leq x$. In addition, it holds that

$$\sum_{(F,\mu)\in\mathcal{C}'} \mu = \sum_{(F,\mu)\in\mathcal{C}-\mathcal{L}} \mu + \sum_{(F,\mu)\in\mathcal{L}} (\mu' + \mu'') = \sum_{(F,\mu)\in\mathcal{C}} \mu = 1,$$

where $\mu' + \mu'' = \mu$ follows from the definition. These indicate that $\mathcal{C}'$ is a fractional forest packing of (V, x). □

6.3 Proof of the fractional set connector packing theorem

First of all, let us review a fractional version of Tutte's tree packing theorem [71].

Theorem 6.4 ([71]). *Let $G = (V, x)$ be an edge-weighted undirected graph. Then there exists a fractional spanning tree packing of G if and only if*

$$x(\delta(\mathcal{P})) \geq |\mathcal{P}| - 1 \quad \textit{for every partition } \mathcal{P} \textit{ of } V \textit{ into non-empty classes.} \tag{6.1}$$

□

We can derive the following lemma from the above theorem.

Lemma 6.3. *Let $G = (V, x)$ be an edge-weighted undirected graph, and $K \subset V$ be an inclusion-wise minimal subset such that $x(\delta(K)) < 2$. Then there exists a fractional spanning tree packing of $(K, x_{\binom{K}{2}})$.*

Proof. We show that (6.1) holds for graph $(K, x_{\binom{K}{2}})$. Let $\mathcal{P}$ be a partition of K into non-empty classes. Then for any $X \in \mathcal{P}$ (i.e., $X \subset K$), it holds that $x(\delta(X)) \geq 2$ by the minimality of K. Therefore

$$x_{\binom{K}{2}}(\delta(\mathcal{P})) = (\sum_{X\in\mathcal{P}} x(\delta(X)) - x(\delta(K)))/2 > |\mathcal{P}| - 1$$

holds. Then by applying Theorem 6.4 to $(K, x_{\binom{K}{2}})$, we can obtain a fractional spanning tree packing of $(K, x_{\binom{K}{2}})$. □

To prove Theorem 6.3, we use a result on the Steiner forest packing due to C. Chekuri and F. B. Shepherd [11]. Here we state a fractional packing version of Theorem 6.1. The proof is based on that of C. Chekuri and F. B. Shepherd [11].

Theorem 6.5. *Let $G = (V, x)$ be an edge-weighted undirected graph. Then there exists a fractional forest packing $\mathcal{C}$ of G such that $\lambda(u, v; G_F) \geq 1$ for every $F \in \mathcal{C}$ and $u, v \in V$ with $\lambda(u, v; V, x) \geq 2$.*

Proof. We prove this theorem by an induction on the number N of 2-edge-connected components in (V, x). First, let us consider the case of $N = 1$. Then for any non-empty $X \subset V$, it holds that $x(\delta(X)) \geq 2$, which implies that (6.1) holds for x because $x(\delta(\mathcal{P})) = \sum_{X\in\mathcal{P}} x(\delta(X))/2 \geq |\mathcal{P}|$. Therefore, we can obtain a required fractional forest packing by Theorem 6.4.

Next, consider the case of $N \geq 2$. Let $K \subset V$ be an inclusion-wise minimal subset such that $x(\delta(K)) < 2$ (such K exists since the edge-connectivity between two vertices in different components is less than 2). Then K is the union of some 2-edge-connected components. By Lemma 6.3, there exists a fractional spanning tree packing $\{(T_i, \pi_i) \mid 1 \leq i \leq p\}$ of $(K, x_{\binom{K}{2}})$. Let $G' = (V' = (V - K), x' \in \binom{V'}{2})$ be the graph obtained by contracting K into a single vertex v_K, executing the complete splitting at v_K, and removing isolated v_K. Note that any two vertices $u, v \in V'$ that belong to the same 2-edge-connected component in (V, x) remains 2-edge-connected in (V', x').

By the inductive hypothesis, (V', x') has a fractional forest packing $\{(H_i, \kappa_i) \mid 1 \leq i \leq q\}$ such that each of $H_1, \ldots, H_q$ connects every two vertices $u, v \in V'$ with $\lambda(u, v; V', x') \geq 2$ (and hence $\lambda(u, v; V, x) \geq 2$). Let $\{(H'_i, \kappa'_i) \mid i = 1, \ldots, q'\}$ be the fractional forest decomposition of (V, x) obtained from $\{(H_i, \kappa_i) \mid i = 1, \ldots, q\}$ by applying Lemmas 6.1 and 6.2. Then $\{(T_i \cup H'_j, \pi_i \kappa'_j) \mid 1 \leq i \leq p, 1 \leq j \leq q'\}$ is a required fractional forest packing. □

Lemma 6.4. *Let $G = (V, x)$ be an edge-weighted undirected graph, and $\mathcal{V}_1, \ldots, \mathcal{V}_m \subseteq 2^V$ be families of disjoint vertex subsets such that $\lambda(\mathcal{V}_i; V, x) \geq 2(\sum_{X \in \mathcal{V}_i} |X| - 1)$ for every $i = 1, \ldots, m$. If $F \subseteq \binom{V}{2}$ satisfies $\lambda(u, v; G_F) \geq 1$ for all $u, v \in V$ with $\lambda(u, v; V, x) \geq 2$, then F is a set connector for $\mathcal{V}_1, \ldots, \mathcal{V}_m$.*

Proof. Let $\mathcal{V}_i \in \{\mathcal{V}_1, \ldots, \mathcal{V}_m\}$. We show that, for any partition $\{\{X_1, \ldots, X_q\}, \{Y_1, \ldots, Y_r\}\}$ of $\mathcal{V}_i$ into two classes, there exists at least one pair $\{u, v\}$ of vertices $u \in \cup_{j=1}^q X_i$ and $v \in \cup_{j=1}^r Y_j$ with $\lambda(u, v; G, x) \geq 2$. This implies the lemma since an edge set that connects such u and v is a set connector in this case.

Now we suppose conversely that $\lambda(u, v; V, x) < 2$ holds for every $u \in \cup_{j=1}^q X_j$ and $v \in \cup_{j=1}^r Y_j$. Then we construct a partition $\mathcal{P}$ of V and a family $\mathcal{Q} \subseteq 2^V$ of vertex subsets as follows. First we set $\mathcal{P} = \{V\}$ and $\mathcal{Q} = \emptyset$. Let us consider the moment at which some two vertices $u \in \cup_{j=1}^q X_j$ and $v \in \cup_{j=1}^r Y_j$ belong to the same class of $\mathcal{P}$. Then choose $W \subset V$ such that $u \in W$, $v \in V - W$ and $x(\delta(W)) < 2$ (such W exists since $\lambda(u, v; G, x) < 2$) and update $\mathcal{P} := \cup_{Z \in \mathcal{P}} \{Z \cap W, Z - W\}$ and $\mathcal{Q} := \mathcal{Q} \cup \{W\}$. Repeat this procedure until every two vertices $u' \in \cup_{j=1}^q X_j$ and $v' \in \cup_{j=1}^r Y_j$ belong to different classes of $\mathcal{P}$.

By the way of construction, it hold that

$$\mathcal{Q}| \leq | \cup_{j=1}^q X_j| + | \cup_{j=1}^r Y_j| - 1 = \sum_{X \in \mathcal{V}_i} |X| - 1$$

and that $\delta(\mathcal{P}) \subseteq \cup_{W \in \mathcal{Q}} \delta(W)$. Now let $U = \cup_{j=1}^p V_j$, where $V_1, \ldots, V_p$ denote the classes of $\mathcal{P}$ that contain vertices in $\cup_{j=1}^q X_j$. Notice that U separates $\mathcal{V}_i$. Since $x(\cup_{W \in \mathcal{Q}} \delta(W)) < 2|\mathcal{Q}| \leq 2(\sum_{X \in \mathcal{V}_i} |X| - 1)$, it holds that

$$x(\delta(U)) \leq x(\cup_{j=1}^p \delta(V_j)) \leq x(\delta(\mathcal{P})) < 2(\sum_{X \in \mathcal{V}_i} |X| - 1).$$

These facts imply that $\lambda(\mathcal{V}_i; V, x) < 2(\sum_{X \in \mathcal{V}_i} |X| - 1)$, a contradiction. □

Now we are ready to prove Theorem 6.3. In the proof, we show the following observation together with Theorem 6.3.

Observation 6.1. *Set connectors in Theorem 6.3 can be given as forests connecting all vertices in each* 2*-edge-connected component of* $(V, 2\alpha x)$. □

Proof of Theorem 6.3 and Observation 6.1. Since $x \in \mathrm{P_{sc}}$, we see that $\lambda(\mathcal{V}_i; V, x) = \min\{x(\delta(X)) \mid X \text{ separates } \mathcal{V}_i\} \geq 1$ holds for every $1 \leq i \leq m$. Therefore, $\lambda(\mathcal{V}_i; G, 2\alpha x) = 2\alpha\lambda(\mathcal{V}_i; G, x) \geq 2\alpha \geq 2(\sum_{X \in \mathcal{V}_i} |X| - 1)$ holds for $1 \leq i \leq m$. By Lemma 6.4, at least one pair $\{u, v\}$ of vertices $u \in \cup_{i=1}^q X_i$ and $v \in \cup_{i=1}^r Y_i$ is contained in the same 2-edge-connected component of $(V, 2\alpha x)$ for any partition $\{\{X_1, \ldots, X_q\}, \{Y_1, \ldots, Y_r\}\}$ of $\mathcal{V}_i$ into two classes. Hence every forest that connects all vertices in each 2-edge-connected component of $(V, 2\alpha x)$ is a set connector.

By Theorem 6.5, there exist a fractional forest packing $\{F_1, \ldots, F_k\}$ of $(V, 2\alpha x)$ such that every two vertices $u, v \in V$ with $\lambda(u, v; V, 2\alpha x) \geq 2$ are connected by each of $F_1, \ldots, F_k$. By the above observation, this is a desired factional set connector packing. □

As a corollary of Theorem 6.3, we can see that the integrality gap of $\mathrm{LP_{sc}}$ is at most 2α.

Corollary 6.1. *For any vectors* $x \in \mathrm{P_{sc}}$ *and* $w \in \mathbb{Q}_+^E$*, there always exists a set connector* $F \subseteq E$ *such that* $2\alpha w^T x \geq w(F)$*. Such* F *can be given as a forest connecting all vertices in each* 2*-edge-connected component of* $(V, 2\alpha x)$.

Proof. By Theorem 6.3, there exists a set connector decomposition $\{(F_i, \mu_i) \mid i = 1, \ldots, k\}$ of $(G, 2\alpha x)$, where $F_1, \ldots, F_k$ can be given as forests connecting all vertices in each 2-edge-connected component of $(V, 2\alpha x)$ by Observation 6.1. Let F_j attain $\min\{w(F_i) \mid i = 1, \ldots, k\}$. Then $2\alpha w^T x \geq \sum_{i=1}^k \mu_i w(F_i) \geq w(F_j)$, as required. □

This gap is tight in the following instance. Given an integer $d \geq 1$, let $G = (V, E)$ be the complete graph on a vertex set V of cardinality $n > 2d$, and $w(e) = 1$ for all $e \in E$. Moreover specify a vertex $s \in V$ and define $\mathcal{V}_1, \ldots, \mathcal{V}_m$ as the families $\{\{s\}, U\}$ for all subsets $U \subseteq V - s$ with $|U| = \alpha$, where $m = \binom{|V|-1}{\alpha}$. In this instance, $\alpha = \max_{1 \leq i \leq m} \sum_{X \in \mathcal{V}_i} |X| - 1$ holds.

Define a rational vector $x \in \mathbb{Q}_+^E$ as $x(e) = 1/(n-1)$ if e is incident to s, and $x(e) = 1/(a(n-1))$ otherwise. Then we can verify that $x \in \mathrm{P_{sc}}$ holds. Hence the optimal cost of rational solutions is at most $w^T x = (n-1)/(n-1) + \binom{n-1}{2}/(\alpha(n-1)) = (n+2\alpha-2)/(2\alpha)$. On the other hand, let us consider an optimal integral solution $F \subseteq E$. Consider the connected component S that contains s in G_F. If $|S| < n - \alpha + 1$, i.e., $|V - S| \geq \alpha$, then $0 = \delta(S; G_F) \geq \lambda(\mathcal{V}_i; G_F)$ would hold for some $\mathcal{V}_i = \{\{s\}, U\}$ with a set $U \subseteq V - S$. Hence $|S| \geq n - \alpha + 1$. By this, $|F| \geq |S| - 1 \geq n - \alpha + 1 - 1 = n - \alpha$. Therefore the integrality gap of this instance is

$$\frac{\text{The optimal cost of integer solutions}}{\text{The optimal cost of rational solutions}} \geq \frac{w(F)}{w^T x} \geq \frac{n - \alpha}{(n + 2\alpha - 2)/(2\alpha)}.$$

We can see that the right-most term approaches 2α as n gets larger.

6.4 Approximation algorithm

In Corollary 6.1, we saw that any vector $x \in \mathrm{P_{sc}}$ can be rounded to a set connector F with $w(F) \leq 2\alpha w^T x$, and that such F can be given as a forest connecting all vertices in each 2-edge-connected component of $(V, 2\alpha x)$ (i.e., $1/\alpha$-edge-connected component of (V, x)). Hence by applying a ρ-approximation algorithm of the Steiner forest problem to constructing such a forest in G, we have a $2\alpha\rho$-approximation algorithm for the set connector problem, where currently $\rho = 2$ is known [31]. However, the arguments in Section 6.3 indicate a 2α-approximation algorithm for the set connector problem. In this section, we describe this.

First, our algorithm computes an optimal solution x of $\mathrm{LP}_{\mathrm{sc}}$ for the given instance consisting of $G = (V, E)$, $w \in \mathbb{Q}_+^E$, and $\mathcal{V}_1, \ldots, \mathcal{V}_m \subseteq 2^V$. We then augment x into $\mathbb{R}_+^{\binom{V}{2}}$ by adding 0's, and w into $\mathbb{Q}_+^{\binom{V}{2}}$ by adding $+\infty$'s if G is not complete. Then our algorithm constructs a forest $F \subseteq E_x \subseteq E$ that connects all vertices in each $1/\alpha$-edge-connected component of (V, x) as follows.

Let $K \subseteq V$ be an inclusion-wise minimal vertex set such that $x(\delta(K)) < 1/\alpha$. Recall that the proof of Theorem 6.5 computes a fractional tree packing of $(K, x_{\binom{K}{2}})$ by applying Lemma 6.3. Instead of this, our algorithm computes a minimum cost tree $T_K \subseteq E_x \cap E[K]$ spanning K. Then we contract K into a single vertex v_K, and execute a complete splitting at v_K. When our algorithm executes contraction or splitting, it modifies the edge cost simultaneously. After this, it recursively computes a sequence of trees in the resulting edge-weighted graph and edge cost until the vertex set becomes a singleton. As reverse operations of contraction and splitting, our algorithm modifies the forest and output the union of T_K and the modified forest as a solution. Below, we describe how to modify the edge cost and how to modify the forest in the reverse operations of contraction and splitting.

First, let us consider the contraction. Let $x' \in \mathbb{R}_+^{\binom{V'}{2}}$ be the vector obtained from x by the contracting K into v_K, where $V' = (V - K) \cup v_K$. Together with this contraction, our algorithm modifies edge cost w into $w' \in \mathbb{Q}_+^{\binom{V'}{2}}$ so that $w'(uv_K) = \min\{w(us) \mid s \in K, x(us) > 0\}$ for each $u \in V - K$ and $w'(uv) = w(uv)$ for each $u, v \in V - K$. Suppose that our algorithm has computed a forest $F' \subseteq E_{x'}$ for (V', x') and w'. Then it constructs a forest $F \subseteq E_x - \binom{K}{2}$ for (V, x) and w from F' in the reverse operation of the contraction as follows. If F' contains no edge in $\delta(v_K)$, we set F to be F'. Otherwise, prepare an edge uv such that $w(uv) = w'(uv_K)$ for each $uv_K \in F' \cap \delta(v_K)$, and let F'' be the set of those edges. Then F is defined as $(F' - \delta(v_K)) \cup F''$. Notice that $w(F) = w'(F')$ holds. Moreover, $F' \cup T_K$ connects every two vertices connected by F.

Next, let us consider the splitting. Let $x' \in \mathbb{R}_+^{\binom{V'}{2}}$ be the vector from $x \in \mathbb{R}_+^{\binom{V'}{2}}$ by splitting a pair $\{v_K a, v_K b\}$ of edges by $\epsilon_{v_K a, v_K b} > 0$ in the complete splitting at v_K. Together with this splitting, our algorithm modifies the edge cost w into a new cost $w' \in \mathbb{Q}_+^{\binom{V'}{2}}$ so that $w'(ab) = \min\{w(ab), w(v_K a) + w(v_K b)\}$ if $x(ab) > 0$ and $w'(ab) = w(v_K a) + w(v_K b)$ otherwise while $w'(e) = w(e)$ for $e \in \binom{V'}{2} - ab$. Suppose that our algorithm has computed a forest $F' \subseteq E_{x'}$ for (V', x') and w'. Then it constructs a forest $F \subseteq E_x$ for (V, x) and w from F' in the reverse operation of the splitting as follows. If $w'(ab) = w(v_K a) + w(v_K b)$, then F is set to $(F' - ab) \cup \{v_K a, v_K b\}$. Otherwise, F is set to be F'. Notice that $w(F) = w'(F')$ holds in both cases.

We note that the reverse operation of contraction and splitting described above can be easily executed by maintaining $p(e)$ for each $e \in \binom{V}{2}$. At the beginning of our algorithm,

$p(e)$ is set to be $\{e\}$. Our algorithm updates $p(uv_K) := p(uv)$ when a set K containing v is contracted into v_K and $w'(uv_K)$ is defined as $w(uv)$, and $p(ab) := p(v_K a) \cup p(v_K b)$ when a pair $\{v_K a, v_K b\}$ is split and $w'(ab)$ is updated to $w(v_K a) + w(v_K b)$. Observe that $\cup_{e\in F'} p(e)$ represents the edge set constructed from a forest F' in both reverse operations.

Now we are ready to see the entire algorithm. The following describes how to compute a solution after an optimal solution x of $\mathrm{LP}_{\mathrm{sc}}$ is given.

Algorithm SETCONNECT

Input: A vertex set V, a vector $x \in \mathbb{R}_+^{\binom{V}{2}}$, and an edge cost $w \in \mathbb{Q}_+^{\binom{V}{2}}$
Output: A forest $F \subseteq \binom{V}{2}$

$K :=$ an inclusion-wise minimal $X \subseteq V$ with $x(\delta(X)) < 1/\alpha$;
Compute a minimum cost tree $T_K \subseteq E_x \cap \binom{K}{2}$ spanning K; # possibly $|K| = 1$ or $K = V$
if $|V| - 1 \leq |K| \leq |V|$ **then**
  Return $F := T_K$ as a solution and halt
end if;
  # contract K into v_K
  $w' := w$;
  For each $e \in \binom{V}{2}$, define $p(e) := \{e\}$;
  $V' := (V - K) \cup v_K$; $x'_{\binom{V-K}{2}} := x_{\binom{V-K}{2}}$;
for $u \in V - K$ **do**
  $x'(uv_K) := \sum_{v\in K} x(uv)$;
  if $x'(uv_K) > 0$ **then**
    $e :=$ an edge attaining $\min\{c(uv) \mid v \in K, x(uv) > 0\}$;
    $w'(uv_K) := w(e)$; $p(uv_K) := p(e)$
  end if
end for;
# complete splitting at v_K
for distinct $a, b \in V' - v_K$ **do**
  Compute $\epsilon_{a,b}$ in (V', x');
  if $\epsilon_{a,b} > 0$ **and** $x'(ab) = 0$ or $w'(ab) > w'(v_K a) + w'(v_K b)$ **then**
    $w'(ab) := w'(v_K a) + w'(v_K b)$; $p(ab) := p(v_K a) \cup p(v_K b)$
  end if;
  $x'(v_K a) := x'(v_K a) - \epsilon_{a,b}$; $x'(v_K b) := x'(v_K b) - \epsilon_{a,b}$; $x'(ab) := x'(ab) + \epsilon_{a,b}$
end for;
$V' := V' - v_K$;
$F' :=$ A solution output by SETCONNECT applied to V', $x'_{\binom{V'}{2}}$ and w';
Return $F := T_K \cup_{e\in F'} p(e)$ as a solution;

Theorem 6.6. *The set connector problem can be approximated within factor of* 2α *by applying algorithm SETCONNECT to an optimal solution* x *of* $\mathrm{LP}_{\mathrm{sc}}$.

Proof. First, let us see that SETCONNECT returns a forest $F \subseteq E_x$ connecting all vertices in a $1/\alpha$-edge-connected component of (V, x) by the induction on $|V|$, the combination of which and Lemma 6.4 implies that F is a set connector for G and $\mathcal{V}_1, \ldots, \mathcal{V}_m$.

By the choice of T_K, it holds that $T_K \subseteq E_x$. By the induction hypothesis, $F' \subseteq E_{x'}$, and then $\cup_{e \in F'} p(e) \subseteq E_x$. Since $F = T_K \cup_{e \in F'} p(e)$, we have $F \subseteq E_x$. On the other hand, let u and v be two vertices in V such that $\lambda(u, v; V, x) \geq 1/\alpha$. Then these are contained in either K or $V - K$ during the algorithm. If $u, v \in K$, these are connected by F since F contains a tree T_K spanning K. In what follows, we suppose that $u, v \in V - K$. Let x' represent the vector maintained in the end of the algorithm. Since contracting K into v_K and the complete splitting at v_K does not decrease the edge-connectivity between u and v, it follows that $\lambda(u, v; V', x'_{\binom{V'}{2}}) \geq 1/\alpha$. By the inductive hypothesis, F' connects u and v, and thereby $F = T_K \cup_{e \in F'} p(e)$ connects such u and v.

Next, let $\{(C_i, \mu_i) \mid i = 1, \ldots, k\}$ be a fractional set connector packing of $(V, 2\alpha x)$ and $\mathcal{V}_1, \ldots, \mathcal{V}_m$ appeared in Theorem 6.3. In the following, we show that $w(F) \leq w(C_i)$ for every $i = 1, \ldots, k$ by the induction on $|V|$ again. This implies that F is a 2α-approximate solution for the set connector problem, as in the proof of Corollary 6.1.

Recall that the proof of Theorem 6.3 constructs C_i as the union of T and $\cup_{e \in H} p(e)$, where $T \subseteq E_x \cap \binom{K}{2}$ is a spanning tree on K and $H \subseteq E_{x'} \cap \binom{V-K}{2}$ is a forest in a fractional forest packing of (V', x'). By the choice of T_K, obviously $w(T_K) \leq w(T)$ holds. On the other hand, $w'(F') \leq w'(H)$ by the inductive hypothesis. As observed in the above, it holds that $w'(F') = w(\cup_{e \in F'} p(e))$ and $w'(H) = w(\cup_{e \in H} p(e))$. Since $F = T_K \cup_{e \in F'} p(e)$ and $C_i = T \cup_{e \in H} p(e)$, we have obtained $w(F) \leq w(C_i)$. □

We note that running time of algorithm SETCONNECT is strongly polynomial, where we use Tardos' algorithm [76] to solve $\mathrm{LP}_{\mathrm{sc}}$. All steps of algorithm SETCONNECT except solving $\mathrm{LP}_{\mathrm{sc}}$ are combinatorial.

6.5 Applications

In this section, we review some problems related to the set connector problem.

6.5.1 NA-connectivity

Here we mention the prior works on the *node to area connectivity* (*NA-connectivity*). H. Ito [42] considered the edge-connectivity $\lambda(v, X)$ between a vertex $v \in V$ and a vertex subset $X \subseteq V$, and called it NA-connectivity. Then augmentation-type problem of NA-connectivity was considered by some researchers [43, 41, 59]. For example, the following problem was shown to be NP-hard by H. Miwa and H. Ito [59].

1-NA-connectivity augmentation problem
Given an undirected graph $G = (V, E)$ and a family $\mathcal{V} \subseteq 2^V$, find an edge set $F \subseteq \binom{V}{2} - E$ of minimum cardinality such that $\lambda(v; X; G_{E \cup F}) \geq 1$ holds for all $X \in \mathcal{V}$ and $v \in V - X$.

By using an algorithm due to Z. Nutov [63], this problem can be approximated within 7/4.

The edge-connectivity for a family of vertex subsets we defined in this paper generalizes the NA-connectivity since $\lambda(v, X; G) = \lambda(\mathcal{V}_X; G)$ holds if we set $\mathcal{V}_X = \{\{v\}, X\}$ for $X \in \mathcal{V}$. Hence the above augmentation problem is contained in the set connector problem even if it is generalized so that an edge cost $c : \binom{V}{2} - E \to \mathbb{Q}_+$ is also given and $c(F)$ is minimized.

Theorem 6.7. *The 1-NA-connectivity augmentation problem with an edge cost can be approximated within a factor of* $2 \max_{X \in \mathcal{V}} |X|$. □

6.5.2 Steiner forest problem

The Steiner forest problem is formulated as follows.

> **Steiner forest problem**
> Given an undirected graph $G = (V, E)$, an edge cost $w : E \to \mathbb{Q}_+$ and disjoint vertex subsets $X_1, \ldots, X_\ell \subseteq V$, find a minimum cost edge set $F \subseteq E$ that connects every two vertices in X_i for every $i = 1, \ldots, \ell$.

The Steiner forest problem can be formulated as the set connector problem by setting each family $\mathcal{V}_i$ of vertex subsets as $\{\{u\}, \{v\}\}$, where $u, v \in X_j, j = 1, \ldots, \ell$. Our algorithm to the set connector problem attains the approximation factor of $2\alpha = 2$, which coincides with the prior best result on the Steiner forest problem [31].

6.5.3 Group Steiner tree problem

The group Steiner tree problem is formulated as follows.

> **Group Steiner tree problem**
> Given an undirected graph $G = (V, E)$, an edge cost $w : E \to \mathbb{Q}_+$, and a family $\mathcal{U} \subseteq 2^V$ of vertex subsets, find a minimum cost tree $T \subseteq E$ which spans at least one vertex in every $X \in \mathcal{U}$.

The group Steiner tree problem is obviously a generalization of the Steiner tree problem. This problem was introduced by G. Reich and P. Widmayer [66]. Their motivation came from the wire routing with multi-port terminals in VLSI design. Afterwards it turned out that this problem has a close relationship with the Steiner tree problem both in undirected graphs and in digraphs [37, 81]. With regards to the approximation hardness, the set cover problem can be reduced to the group Steiner tree problem ([15] presents a reduction from the set cover problem to the terminal Steiner tree problem, which is a special case of the group Steiner tree problem as described below). By this fact and the approximation hardness of the set cover problem described in Theorem 2.7, the group Steiner tree problem does not admit any approximation factor of $(1-\epsilon)\ln m$ unless NP $\subset$ DTIME$(n^{\log\log n})$, where $\epsilon > 0$, $m = |\mathcal{U}|$ and $n = |\cup_{X \in \mathcal{U}} X|$. Besides this, E. Halperin and R. Krauthgamer [38] proved that the group

Steiner tree problem is hard to approximate within a factor better than $\Omega(\log^{2-\epsilon} m)$ for every $\epsilon > 0$ unless NP problems have quasi-polynomial time Las-Vegas algorithms. On the other hand, a $(1 + \ln m/2)\sqrt{m}$-approximation algorithm was proposed by C. D. Bateman et al. [4]. Currently the best approximation factors are $O(\log m \log |V| \log N)$ due to [9, 10, 29], and $2N$ due to P. Slavík [72], where $N = \max_{X \in \mathcal{U}} |X|$.

Although the set connector problem resembles the group Steiner tree problem, they are different in the fact that the set connectors may be forests. However, the group Steiner tree problem can be reduced to the set connector problem as follows. Pick up a designated subset $S \in \mathcal{U}$. For each $s \in S$, run the algorithm of the set connector problem for the instance with G, c, and $\mathcal{V}_U = \{s, U\}$, $U \in \mathcal{U} - S$. Then this provides the approximation factor of $2\alpha = 2N$. This approximation factor coincides with Slavík's result [72].

Theorem 6.8. *The group Steiner tree problem can be approximated within a factor of* $2 \max_{X \in \mathcal{V}} |X|$. □

6.5.4 Tree cover problem

As mentioned in the above, the group Steiner tree problem contains the Steiner tree problem and the set cover problem. In addition to these, it also contains some fundamental optimization problems. In the following two subsections, we review two examples of them. Here we see the tree cover problem, which is formulated as follows.

Tree cover problem
Given a simple undirected graph $G = (V, E)$, an edge cost $w : E \to \mathbb{Q}_+$, and a set $F \subseteq E$ of edges, find a minimum cost tree $T \subseteq E$ such that each edge in F is contained in T or shares one end vertex with some edge in T.

A tree cover can be also regarded as a connected edge dominating set (see Chapter 3). This problem was introduced by E. M. Arkin et al. [3], where they considered only the setting of $F = E$. Their motivation is to locate tree-shaped facilities on a graph. With regards to the algorithmic results, a 3.55-approximation algorithm was proposed by them [3]. Afterwards, the approximation factor was improved to 3 by J. Könemann et al. [50] and by T. Fujito [24]. Furthermore this was improved to 2 by T. Fujito [25]. These algorithms can also deal with only the case of $F = E$.

On the other hand, the reduction to the set connector problem can approximate the tree cover problem as follows. The advantage of this reduction is the fact that it can approximate the instance also with $F \subset E$.

Theorem 6.9. *The tree cover problem can be approximated within* 4.

Proof. First, designate an edge $f \in F$. For each end vertex s of f, execute the algorithm SETCONNECT to the instance with G, c, and $\mathcal{V}_e$ $(e \in F - f)$, where $\mathcal{V}_e = \{s, \{v_e, v'_e\}\}$, and v_e and v'_e are the end vertices of edge e. Then a solution attaining the minimum cost in the two executions is an approximate solution for the tree cover problem. The approximation factor is $2\alpha = 2 \max_{e \in F-f}(\sum_{X \in \mathcal{V}_e} |X| - 1) = 4$. □

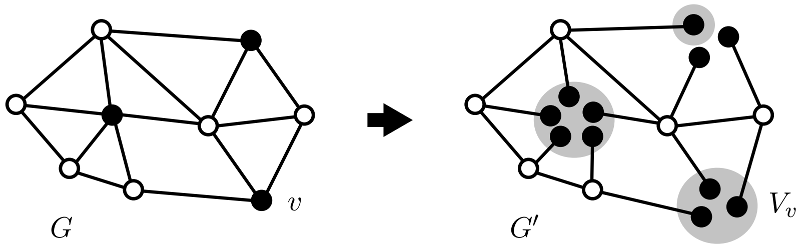


Figure 6.2: Reduction from the terminal Steiner tree problem to the set connector problem

6.5.5 Terminal Steiner tree problem

The *terminal Steiner tree problem* (which some researchers call *full Steiner tree problem*) is formulated as follows.

> **Terminal Steiner tree problem**
> Given a simple undirected graph $G = (V, E)$, an edge cost $w : E \to \mathbb{Q}_+$, and a set $U \subseteq V$ of terminals, find a minimum cost terminal Steiner tree, which is defined as a tree $T \subseteq E$ in which terminals in U are leafs (i.e., the degree of every terminal is one).

This problem is introduced by G. Lin and G. Xue [53] as a variant of the Steiner tree problem, noting the application to designing VLSI and telecommunication network. With regards to the algorithms, G. Lin and G. Xue [53] proposed a $(2+\rho)$-approximation algorithm and D. E. Drake and S. Hougardy [15] proposed a 2ρ-approximation algorithm, both assuming the metric edge cost, where ρ denotes the best approximation factor for the Steiner tree problem (currently $\rho = 1 + (\ln 3)/2 \leq 1.55$ is known [67]). Moreover C. L. Lu et al. [56] presented an 8/5-approximation algorithm for the instances in which the edge costs are restricted to either 1 or 2 on a complete graph. They also showed that those instances are NP-complete and MAX SNP-hard. An application of their algorithm to the reconstruction of evolutionary tree in biology was also proposed. For a general edge cost, it is shown that no polynomial time algorithm can approximate the terminal Steiner tree problem within a factor of $(1-\epsilon)\ln n$ unless NP $\subset$ DTIME$(n^{O(\log\log n)})$ [15] by presenting a reduction from the set cover problem, where $n = |U|$.

Our approximation algorithm for the set connector problem gives an approximation algorithm for the terminal Steiner tree problem with general edge cost. Its approximation factor is $2\max_{v\in U}|\delta(v; E)|$.

Theorem 6.10. *The terminal Steiner tree problem can be approximated within a factor of* $2\max_{v\in U}|\delta(v; E)|$.

Proof. Let G' denote the graph obtained from G by replacing each vertex $v \in U$ by a set $V_v = \{v_1, \ldots, v_{|\delta(v;E)|}\}$ of new vertices of degree 1 so that each $v_i \in V_v$ will be a new end vertex of each edge incident to v (see Figure 6.2). We designate a vertex $s \in U$. For each $s_i \in V_s$, consider an instance I_{s_i} of the set connector problem G', c, and $\mathcal{V}_v = \{\{s_i\}, V_v\}$

$(v \in Y - s)$, and compute a set connector to I_{s_i} by SETCONNECT. Then a set connector F attaining the minimum cost among instances I_{s_i} is a tree cover of the approximation factor $2\alpha \leq 2\max_{v \in U} |\delta(v; E)|$. □

Chapter 7

Conclusion

In this thesis, we discussed the approximability of some network design problems. Here we would like to mention the remaining problems and to suggest possible future works in order to stimulate further investigation.

In Chapter 3, we considered some variants of the edge dominating set problem and the edge cover problem. For example, we proposed an approximation algorithm to the capacitated b-edge dominating set problem, whose approximation factor is estimated in terms of a vector in $\mathrm{EDS}(G, b, c)$. A natural way to improve this algorithm is to use a more refined polyhedron instead of $\mathrm{EDS}(G, b, c)$. Actually, 2-approximation algorithms for the edge dominating set problem [26, 65] uses a polyhedron obtained by adding some vertex cover type inequalities to $\mathrm{EDS}(G, 1, +\infty)$. However, A. Berger et al. [6] showed that adding the type of inequalities is hard to improve our algorithm.

In Chapter 4, we presented some theorems on the existence of strongly splittable pairs and detachments preserving the local edge-connectivity. However, a condition for a graph to have an r-edge-connected detachment remains open yet in general. It is also interesting to show a condition for such a detachment to have some further property as we considered the existence of loops in Chapter 4.

In Chapter 5, we considered a problem with the edge-connectivity constraints and degree constraints. Although our algorithm solves most cases of this problem, the others still remain unsolved. Hence further development of the algorithm is expected. We think that improving the results in Chapter 4 is important for this. Moreover, modifying the Jain's algorithm may be helpful since our algorithm utilizes the Jain's algorithm for constructing an r-edge-connected graph. We also considered (m, n)-VRP and showed that it is approximable within $1.5 + (m - n)/m$. It is also interesting to generalize (m, n)-VRP by specifying the number of cycles containing each vertex except the depot.

In Chapter 6, we designed a 2α-approximation algorithm for the set connector problem, where $\alpha = \max_{1\leq i\leq m}(\sum_{X\in\mathcal{V}_i} |X|) - 1$. Although we gave an example for the tightness of the integrality gap, each of $\mathcal{V}_i$ consists of two vertex subsets one of which is a singleton in the example. Hence it does not deny that our algorithm would achieve the approximation factor of $\max\{|X| \mid X \in \mathcal{V}_i, 1 \leq i \leq m\}$. It is also interesting to consider a combinatorial approximation algorithm. Furthermore, it remains open to find a least cost subgraph in which families of vertex subsets are highly connected.

Bibliography

[1] R. P. Anstee, *A polynomial algorithm for b-matchings: an alternative approach*, Information Processing Letters, 24 (1987), pp. 153–157.

[2] D. Applegate, R. Bixby, V. Chvátal, and W. Cook, *Traveling salesman problem*, http://www.tsp.gatech.edu/.

[3] E. M. Arkin, M. M. Halldórsson, and R. Hassin, *Approximating the tree and tour covers of a graph*, Information Processing Letters, 47 (1993), pp. 275–282.

[4] C. D. Bateman, C. S. Helvig, G. Robins, and A. Zelikovsky, *Provably good routing tree construction with multi-port terminals*, in Proceedings of the 1997 International Symposium on Physical Design, 1997, pp. 96–102.

[5] A. R. Berg, B. Jackson, and T. Jordán, *Highly edge-connected detachments of graphs and digraphs*, Journal of Graph Theory, 43 (2003), pp. 67–77.

[6] A. Berger, T. Fukunaga, H. Nagamochi, and O. Parekh, *Capacitated b-edge dominating set and related problems*, submitted.

[7] K. Cameron, *Induced matchings*, Discrete Applied Mathematics, 24 (1989), pp. 97–102.

[8] R. Carr, T. Fujito, G. Konjevod, and O. Parekh, *A $2\frac{1}{10}$-approximation algorithm for a generalization of the weighted edge-dominating set problem*, Journal of Combinatorial Optimization, 5 (2001), pp. 317–326.

[9] M. Charikar, C. Chekuri, A. Goel, and S. Guha, *Rounding via trees: deterministic approximation algorithms for group Steiner trees and k-median*, in Proceedings of the Thirtieth Annual ACM Symposium on Theory of Computing, 1998, pp. 114–123.

[10] C. Chekuri, G. Even, and G. Kortsarzc, *A greedy approximation algorithm for the group Steiner problem*, Discrete Applied Mathematics, 154 (2006), pp. 15–34.

[11] C. Chekuri and F. B. Shepherd, *Approximate integer decompositions for undirected network design problems.* Manuscript, 2004.

[12] M. Chlebík and J. Chlebíková, *Approximation hardness of minimum edge dominating set and minimum maximal matching*, in Proceedings of the 14th Annual International Symposium on Algorithms and Computation, 2003, pp. 415–424.

[13] V. Chvátal, *A greedy heuristics for the set covering problem*, Mathematics of Operations Research, 4 (1979), pp. 233–235.

[14] I. Dirnur and S. Safra, *On the importance of being biased*, in Proceedings of the 34th Annual ACM Symposium on Theory of Computing, 2002, pp. 33–42.

[15] D. E. Drake and S. Hougardy, *On approximation algorithms for the terminal Steiner tree problem*, Information Processing Letters, 89 (2004), pp. 15–18.

[16] W. Duckworth, D. F. Manlove, and M. Zito, *On the approximability of the maximum induced matching problem*, Journal of Discrete Algorithms, 3 (2005), pp. 79–91.

[17] U. Feige, *A threshold of* $\ln n$ *for approximating set cover*, Journal of the ACM, 45 (1998), pp. 634–652.

[18] S. P. Fekete, S. Khuller, M. Klemmstein, B. Raghavachari, and N. Young, *A network-flow technique for finding low-weight bounded-degree spanning trees*, Journal of Algorithms, 24 (1997), pp. 310–324.

[19] B. Fleiner, *Detachment of vertices of graphs preserving edge-connectivity*, SIAM Journal on Discrete Mathematics, 18 (2005), pp. 581–591.

[20] A. Frank, *On connectivity properties of Eulerian digraphs*, Annals of Discrete Mathematics, 41 (1989), pp. 179–194.

[21] A. Frank, *Augmenting graphs to meet edge-connectivity requirements*, SIAM Journal on Discrete Mathematics, 5 (1992), pp. 25–53.

[22] A. Frank, *On a theorem of Mader*, Discrete Mathematics, 191 (1992), pp. 49–57.

[23] G. N. Frederickson, M. S. Hecht, and C. E. Kim, *Approximation algorithms for some routing problems*, SIAM Journal of Computing, 7 (1978), pp. 178–193.

[24] T. Fujito, *On approximability of the independent/connected edge dominating set problems*, Information Processing Letters, 79 (2001), pp. 261–266.

[25] T. Fujito, *How to trim an MST: A 2-approximation algorithm for minimum cost tree cover*, in Proceedings of Automata, Languages and Programming, 33rd International Colloquium, vol. 4051 of Lecture Notes in Computer Science, Venice, Italy, 2006, pp. 431–442.

[26] T. Fujito and H. Nagamochi, *A 2-approximation algorithm for the minimum weight edge dominating set problem*, Discrete Applied Mathematics, 118 (2002), pp. 199–207.

[27] T. Fukunaga and H. Nagamochi, *Approximating a generalization of metric TSP*, IEICE Transactions on Information and Systems, Special Section on Foundations of Computer Science, (to appear).

[28] M. R. Garey and D. S. Johnson, *Computers and intractability; a guide to the theory of NP-completeness*, W. H. Freeman & Co., 1979.

[29] N. Garg, G. Konjevod, and R. Ravi, *A polylogarithmic approximation algorithm for the group Steiner tree problem*, Journal of Algorithms, 37 (2000), pp. 66–84.

[30] M. X. Goemans and D. J. Bertsimas, *Survivable networks, linear programming relaxations and the parsimonious property*, Mathematical Programming, 60 (1993), pp. 145–166.

[31] M. X. Goemans and D. P. Williamson, *A general approximation technique for constrained forest problems*, SIAM Journal on Computing, 24 (1995), pp. 296–317.

[32] M. X. Goemans and D. P. Williamson, *The primal-dual method for approximation algorithms and its application to network design problems*, PWS, 1997, ch. 4, pp. 144–191.

[33] M. C. Golumbic and R. C. Laskar, *Irredundancy in circular arc graphs*, Discrete Applied Mathematics, 44 (1993), pp. 79–89.

[34] M. C. Golumbic and M. Lewenstein, *New results on induced matchings*, Discrete Applied Mathematics, 101 (2000), pp. 157–165.

[35] T. Grünwald, *Ein neuer beweis eines mengerschen satzes*, The Journal of the London Mathematical Society, 13 (1938), pp. 188–192.

[36] D. Gusfield, *Connectivity and edge-disjoint spanning trees*, Information Processing Letters, 16 (1983), pp. 87–89.

[37] E. Halperin, G. Kortsarz, R. Krauthgamer, A. Srinivasan, and N. Wang, *Integrality ratio for group Steiner trees and directed steiner trees*, in Proceedings of the fourteenth annual ACM-SIAM symposium on Discrete algorithms, 2003, pp. 275–284.

[38] E. Halperin and R. Krauthgamer, *Polylogarithmic inapproximability*, in Proceedings of the 35th Annual ACM Symposium on Theory of Computing, 2003, pp. 585–594.

[39] F. Harary, *Graph Theory*, Addison-Wesley, 1969.

[40] J. Håstad, *Clique is hard to approximate within* $n^{1-\varepsilon}$, Acta Mathematica, 182 (1999), pp. 105–142.

[41] T. Ishii and M. Hagiwara, *Augmenting local edge-connectivity between vertices and vertex subsets in undirected graphs*, in Proceedings of 28th International Symposium on Mathematical Foundations of Computer Science, vol. 2747 of Lecture Notes in Computer Science, 2003, pp. 490–499.

[42] H. Ito, *Node-to-area connectivity of graphs*, Transactions of the Institute of Electrical Engineers of Japan, 11C (1994), pp. 463–469.

[43] H. Ito and M. Yokoyama, *Edge connectivity between nodes and node-subsets*, Networks, 31 (1998), pp. 157–164.

[44] B. Jackson, *Some remarks on arc-connectivity, vertex splitting, and orientation in digraphs*, Journal of Graph Theory, 12 (1988), pp. 429–436.

[45] K. Jain, *A factor 2 approximation algorithm for the generalized Steiner network problem*, Combinatorica, 21 (2001), pp. 39–60.

[46] D. S. Johnson, *Approximation algorithms for combinatorial problems*, Journal of Computer and System Sciences, 9 (1974), pp. 256–278.

[47] T. Jordán and Z. Szigeti, *Detachments preserving local edge-connectivity of graphs*, SIAM Journal on Discrete Mathematics, 17 (2003), pp. 72–87.

[48] H. Kaplan, M. Lewenstein, N. Shafrir, and M. Sviridenko, *Approximation algorithms for asymmetric TSP by decomposing directed regular multigraphs*, Journal of the ACM, 52 (2005), pp. 602–626.

[49] D. Kobler and U. Rotics, *Finding maximum induced matchings in subclasses of clawfree and* p_5*-free graphs, and in graphs with matching and induced matching of equal maximum size*, Algorithmica, 37 (2003), pp. 327–346.

[50] J. Könemann, G. Konjevod, O. Parekh, and A. Sinha, *Improved approximations for tour and tree covers*, Algorithmica, 38 (2004), pp. 441–449.

[51] B. Korte and J. Vygen, *Combinatorial Optimization: Theory and Algorithms*, Springer, 2000.

[52] E. L. Lawler, J. K. Lenstra, A. H. G. Rinnooy Kan, and D. B. Shmoys, eds., *The Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization*, John Wiley & Sons, 1985.

[53] G. Lin and G. Xue, *On the terminal Steiner tree problem*, Information Processing Letters, 84 (2002), pp. 103–107.

[54] L. Lovász, *On the ratio of optimal integral and fractional covers*, Discrete Mathematics, 13 (1975), pp. 383–390.

[55] V. V. Lozin, *On maximum induced matchings in bipartite graphs*, Information Processing Letters, 81 (2002), pp. 7–11.

[56] C. L. Lu, C. Y. Tang, and R. C.-T. Lee, *The full Steiner tree problem*, Theoretical Computer Science, 306 (2003), pp. 55–67.

[57] W. Mader, *A reduction method for edge-connectivity in graphs*, Annals of Discrete Mathematics, 3 (1978), pp. 145–164.

[58] K. Menger, *Zur allgemeinen kurventheorie*, Fundamenta Mathematicae, 10 (1927), pp. 96–115.

[59] H. Miwa and H. Ito, *Edge augmenting problems for increasing connectivity between vertices and vertex subsets*, in 1999 Technical Report of IPSJ, vol. 99-AL-66, 1999, pp. 17–24.

[60] K. G. Murty and C. Perin, *A 1-matching blossom-type algorithm for edge covering problems*, Networks, 12 (1982), pp. 379–391.

[61] H. Nagamochi, *A detachment algorithm for inferring a graph from path frequency*, in Proceedings of the 12th Annual International Computing and Combinatorics Conference, vol. 411 of Lecture Notes in Computer Science, 2006, pp. 274–283.

[62] C. St. J. A. Nash-Williams, *Connected detachments of graphs and generalized Euler trails*, Journal of London Mathematical Society, 31 (1985), pp. 17–29.

[63] Z. Nutov, *Approximating connectivity augmentation problems*, in Proceedings of the 16th Annual ACM-SIAM Symposium on Discrete Algorithms, 2005, pp. 176–185.

[64] C. H. Papadimitriou and S. Vempala, *On the approximability of the traveling salesman problem*, Combinatorica, 26 (2006), pp. 101–120.

[65] O. Parekh, *Edge dominating and hypomatchable sets*, in Proceedings of the 13th Annual ACM-SIAM Symposium on Discrete Algorithms, 2002, pp. 287–291.

[66] G. Reich and P. Widmayer, *Beyond Steiner's problem: a VLSI oriented generalization*, in Proceedings of Graph-Theoretic Concepts in Computer Science, vol. 411 of Lecture Notes in Computer Science, 1990, pp. 196–210.

[67] G. Robins and A. Zelikovsky, *Improved Steiner tree approximation in graphs*, in Proceedings of 11th Annual ACM-SIAM Symposium on Discrete Algorithms, 2000, pp. 770–779.

[68] S. K. Sahni and T. F. Gonzalez, *P-complete approximation algorithms*, Journal of ACM, 23 (1976), pp. 555–565.

[69] M. R. Salavatipour, *A polynomial time algorithm for strong edge coloring of partial k-trees*, Discrete Applied Mathematics, 143 (2004), pp. 285–291.

[70] A. Schrijver, *Theory of linear and integer programming*, John Wiley & Sons, Inc., 1986.

[71] A. Schrijver, *Combinatorial Optimization: Polyhedra and Efficiency*, Springer, 2003.

[72] P. Slavík, *Approximation algorithms for set cover and related problems*, PhD thesis, University of New York, 1998.

[73] A. Srinivasan, K. Madhukar, P. Nagavamsi, C. P. Rangan, and M.-S. Chang, *Edge domination on bipartite permutation graphs and contriangulated graphs*, Information Processing Letters, 56 (1995), pp. 165–171.

[74] L. J. Stockmeyer and V. V. Vazirani, *NP-completeness of some generalizations of the maximum matching problem*, Information Processing Letters, 15 (1982), pp. 14–19.

[75] É. Tardos, *A strongly polynomial minimum cost circulation algorithm*, Combinatorica, 5 (1985), pp. 247–255.

[76] É. Tardos, *A strongly polynomial algorithm to solve combinatorial linear programs*, Operations Research, 34 (1986), pp. 250–256.

[77] V. Vazirani, *Approximation Algorithm*, Springer, 2001.

[78] L. A. Wolsey, *Heuristic analysis, linear programming and branch and bound*, Mathematical Programming Study, 13 (1980), pp. 121–134.

[79] M. Yannakakis and F. Gavril, *Edge dominating sets in graphs*, SIAM Journal on Applied Mathematics, 38 (1980), pp. 364–372.

[80] M. Zito, *Maximum induced matchings in regular graphs and trees*, in Graph-Theoretic Concepts in Computer Science, 25th International Workshop, vol. 1665 of Lecture Notes in Computer Science, 1999, pp. 89–101.

[81] L. Zosin and S. Khuller, *On directed Steiner trees*, in Proceedings of the 13th Annual ACM-SIAM Symposium on Discrete Algorithms, 2002, pp. 59–63.

List of publications

Journals

1. T. Fukunaga and T. Ibaraki, *Knowledge based support vector machines*, International Journal of Engineering Intelligent Systems for Electrical Engineering and Communications, Special Issue on Knowledge Engineering, 13 (2005), pp. 259–267.

2. T. Fukunaga and H. Nagamochi, *Approximating a generalization of metric TSP*, IEICE Transactions on Information and Systems, Special Section on Foundations of Computer Science (to appear).

3. T. Fukunaga and H. Nagamochi, *Generalizing the induced matching by edge capacity constraints*, Discrete Optimization (to appear).

4. A. Berger, T. Fukunaga, H. Nagamochi, and O. Parekh, *Capacitated b-edge dominating set and related problems* (submitted).

5. T. Fukunaga and H. Nagamochi, *Eulerian detachments with local-edge-connectivity* (submitted).

6. T. Fukunaga and H. Nagamochi, *Approximating minimum cost multigraphs of specified edge-connectivity under degree bounds* (submitted).

7. T. Fukunaga and H. Nagamochi, *Network design with edge-connectivity and degree constraints* (submitted).

Conferences

1. T. Fukunaga and T. Ibaraki, *Knowledge based SVM*, in Proceedings of the International Conference on Knowledge Engineering and Decision Support, Porto, Portugal, July 21-23, 2004, pp. 135–142.

2. T. Fukunaga and H. Nagamochi, *Edge packing problem with edge capacity constraints*, in Proceedings of the 4th Japanese-Hungarian Symposium on Discrete Mathematics and Its Applications, Budapest, Hungary, June 3-6, 2005, pp. 69–75.

3. T. Fukunaga and H. Nagamochi, *Approximation algorithms for the b-edge dominating set problem and its related problems*, in Proceedings of the 11th International Comput-

ing and Combinatorics Conference, vol. 3595 of Lecture Notes in Computer Science, Kunming, China, August 16-19, 2005, pp. 747–756.

4. T. Fukunaga and H. Nagamochi, *Approximating minimum cost multigraphs of specified edge-connectivity under degree bounds*, in Proceedings of the 9th Japan-Korea Joint Workshop on Algorithms and Computation, Sapporo, Japan, July 4-5, 2006, pp. 25–32.

5. T. Fukunaga and H. Nagamochi, *Some theorems on detachments preserving local-edge-connectivity*, in Proceedings of the 5th Cracow Conference on Graph Theory, vol. 24 of Electronic Notes in Discrete Mathematics, Ustron, Poland, September 11-15, 2006, pp. 173–180.

6. T. Fukunaga and H. Nagamochi, *Network design with edge-connectivity and degree constraints*, in Proceedings of the 4th Workshop on Approximation and Online Algorithms, Lecture Notes in Computer Science, Zürich, Switzerland, September 14-15, 2006 (to appear).